**Financial Report Mover – Utility Program to Move Report Templates Between Orgs**

**Overview**

This is a spec for a pair of utility programs (called programs 1 and 2 below) whose purpose is to move financial report templates between orgs. Report templates can’t be copied from org to org with data loader because of (1) the master/detail relationships between the nine template tables and (2) the large number of lookup fields involved.

This process will fill two slightly different needs:

1. Internally for Rootstock, it will allow demo and test templates to be copied from one org to another.
2. For new customers, it will allow templates that were created and tested in a customer’s sandbox to be copied to the production org at go-live.

The process is intended for the use of Rootstock only. It is not a general-purpose program that can be entrusted to users because it is unsafe for amateur use in a few different ways:

1. It assumes that either (1) the chart of accounts in the source org will be the same as the chart of accounts in the destination or (2) that the user company will not copy report sublines (**GLFSTMPLT5** and **GLFSTMPLT8** records), which do the actual work of digging up data, from org to org but will instead recreate them manually.
2. It assumes that all external data (**GLFSTMPLT6**) from the source org can safely overwrite any existing external data in the destination org.
3. It requires the user to manually do any moving of flat files that is required.
4. Program 1 will kill any flat files that already exist when it kicks off.
5. It does not permit the kind of selectivity users will expect (and complain about if they don’t get) in a production process. For instance, it transfers either a single template or all templates for a company – nothing in between.
6. It stops ungracefully if it ever fails to find expected **GLACCT**, **GLCMP**, **GLDIV**, **GLPROJ**, or **GLFSTMPLT6** records in the destination org.
7. It stops ungracefully if it tries to load a template that already exists in the destination org.
8. It assumes a single user environment, in that Program 1 will always initialize the parameter-passing tables (which would devastate any other instance of Program 1 that is already being run by another user).

It is conceivable that it might someday be enhanced so that it’s robust enough for general release – but I wouldn’t be surprised if the resulting “safe” version turned out to be five times the size of the Rootstock-only version.

Program 1, which is run in the source org, will get parameters from the source org user, will initialize table **TMPLTMVR** with those parameters, will initialize table **TMPLTMVRID** with the source-org values of all Salesforce record ID fields (lookups and master/detail) in **GLFSTMPLT1** through **GLFSTMPLT9**, and will dump **TMPLTMVR**, **TMPLTMVRID**, and **GLFSTMPLT1** through **GLFSTMPLT9** to multiple flat files: one file per table being moved plus two (**TMPLTMVR** and **TMPLTMVRID**) that control the process and one (**TMPLTCOA**) the serves as a map of the record IDs of the chart of accounts (**GLACCT**) in the source org.

Program 2, which is run in the destination org, will

1. Create temporary tables **TMPLTMVR**, **TMPLTMVRID**, **TMPLTCOA**, and **TEMPTMPLT1** through **TEMPTMPLT9**.
2. Populate **TMPLTMVR** from the flat file(s) and then read its data.
3. Populate **TMPLTMVRID** from the flat file(s).
4. Populate from **TMPLTCOA** from the flat file(s) and then read the destination org’s **GLACCT** for eachName field in **TMPLTCOA**. If a **TMPLTCOA** Name exists in **GLACCT**, the ID of the **GLACCT** record is written back to **TEMPCOA**.
5. Populate each **TEMPTMPLT*n*** table from the corresponding **GLFSTMPLT*n*** data in the flat file(s). Each **TEMPTMPLT*n*** table is a mirror image of the corresponding **GLFSTMPLT*n*** table, except that the Salesforce record ID fields in the **TEMPTMPLT*n*** table are stored as simple text fields.
6. Build the **GLFSTMPLT*n*** tables from the **TEMPTMPLT*n*** tables in this order:
   1. **GLFSTMPLT6** – Financial statement template table # 6 – external data.
   2. **GLFSTMPLT7** – Financial statement template table # 7 – predefined subline group header.
   3. **GLFSTMPLT8** – Financial statement template table # 8 – predefined subline group detail.
   4. **GLFSTMPLT1** – Financial statement template table # 1 – top-level header.
   5. **GLFSTMPLT2** – Financial statement template table # 2 – variant header.
   6. **GLFSTMPLT3** – Financial statement template table # 3 – column definitions.
   7. **GLFSTMPLT4** – Financial statement template table # 4 – statement line header.
   8. **GLFSTMPLT9** – Financial statement template table # 9 – export label list for enforcing uniqueness.
   9. **GLFSTMPLT5** – Financial statement template table # 5 – statement line calculation detail.

As each row of each temp table is inserted into its permanent version, its record ID will be retrieved and written to **TMPLTMVRID**. **TMPLTMVRID** will thus contain both the source org and destination org values for every important non-chart of accounts record ID. In that way, when lower-level tables are copied from **TEMPTMPLT*n*** to **GLFSTMPLT*n***, the IDs of their master levels and the local values of their lookup fields can be filled in.

**Table schemas**

**Table 1 – GLFSTMPLT1** – Financial statement template table # 1 – top-level header . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Page 4

**Table 2 – GLFSTMPLT2** – Financial statement template table # 2 – variant header . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Page 6

**Table 3 – GLFSTMPLT3** – Financial statement template table # 3 – column definitions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Page 12

**Table 4 – GLFSTMPLT4** – Financial statement template table # 4 – statement line header . . . . . . . . . . . . . . . . . . . . . . . . . . . . Page 25

**Table 5 – GLFSTMPLT5** – Financial statement template table # 5 – statement line calculation detail . . . . . . . . . . . . . . . . . . . Page 37

**Table 6 – GLFSTMPLT6** – Financial statement template table # 6 – external data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Page 40

**Table 7 – GLFSTMPLT7** – Financial statement template table # 7 – predefined subline group header . . . . . . . . . . . . . . . . . . . Page 41

**Table 8 – GLFSTMPLT8** – Financial statement template table # 8 – predefined subline group detail . . . . . . . . . . . . . . . . . . . . Page 41

**Table 9 – GLFSTMPLT9** – Financial statement template table # 9 – export label list for enforcing uniqueness. . . . . . . . . . . . . Page 43

**Table 10** – **TMPLTMVR –** Temp table that preserves the user-entered parameters from the source org . . . . . . . . . . . . . . . . . Page 43

**Table 11** – **TMPLTMVRID –** Temp table that maps source org Salesforce record IDs to destination org Salesforce record IDs Page 43

**Table 12 – TMPLTCOA** – Source org chart of accounts to destination org chart of accounts translation table . . . . . . . . . . . . . Page 44

**Table 13 – TEMPTMPLT1** – Temporary version of **GLFSTMPLT1** – top-level header . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Page 44

**Table 14 – TEMPTMPLT2** – Temporary version of **GLFSTMPLT2** – variant (column definition) header . . . . . . . . . . . . . . . . . . Page 46

**Table 15 – TEMPTMPLT3** – Temporary version of **GLFSTMPLT3** – column definitions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Page 52

**Table 16 – TEMPTMPLT4** – Temporary version of **GLFSTMPLT4** – statement line header . . . . . . . . . . . . . . . . . . . . .. . . . . . . .. Page 58

**Table 17 – TEMPTMPLT5** – Temporary version of **GLFSTMPLT5** – statement line calculation detail . . . . . . . . . . . . . . . . . . . . Page 65

**Table 18 – TEMPTMPLT6** – Temporary version of **GLFSTMPLT6** – external data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Page 67

**Table 19 – TEMPTMPLT7** – Temporary version of **GLFSTMPLT7** – predefined subline group header . . . . . . . . . . . . . . . . . . . . Page 67

**Table 20 – TEMPTMPLT8** – Temporary version of **GLFSTMPLT8** – predefined subline group detail . . . . . . . . . . . . . . . . . . . . . Page 68

**Table 21 – TEMPTMPLT9** – Temporary version of **GLFSTMPLT9** – export label list for enforcing uniqueness. . . . . . . . . . . . . Page 68

**Process Spec** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Page 69

**Screens** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Page 69

**Record ID translation** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Page 70

**Pseudo-code** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Page 83

Table **GLFSTMPLT1** – Financial statement template table # 1: header table for financial statement template.

| **Field label** | **Column** | **Type/size** | **Possible values** | **Meaning / comment** |
| --- | --- | --- | --- | --- |
| **Reporting company** | **GLFSTMPLT1\_CMPNO** | **Lookup to GLCMP** | **-** | **This is the reporting company. If the reporting company is a parent and this is a consolidated statement, the reporting company can be overridden as the source of data at the line calculation level (GLFSTMPLT5).** |
| **Template ID** | **GLFSTMPLT1\_TEMPLATE** | **Text(30)** | **-** | **Identifies the report being specified by the template.** |
| **Report type** | **GLFSTMPLT1\_REPORTTYPE** | **Picklist** | **Balance sheet** | **Formal balance sheet** |
| **Balance sheet schedule** | **Supporting schedule to balance sheet** |
| **P & L** | **Formal P & L** |
| **P & L schedule** | **Supporting schedule to P & L** |
| **Cash flow statement** | **Cash flow statement (aka statement of changes to financial position). Not supported in initial release.** |
| **Cash flow stmt schedule** | **Supporting schedule to cash flow statement. Not supported in initial release.** |
| **Report scope** | **GLFSTMPLT1\_SCOPE** | **Picklist** | **Company wide Single division Single project Company plus dimension Division plus dimension Project plus dimension** | **Must be Company wide if GLFSTMPLT1\_REPORTTYPE = Balance sheet. Must be Company wide or Company plus dimension if GLFSTMPLT1\_REPORTTYPE = Balance sheet schedule.** |
| **Define a constant division for filtering** | **GLFSTMPLT1\_CONSTANTDIV** | **Checkbox** | **-** | **NULL unless GLFSTMPLT1\_SCOPE = Single division, Single project, Division plus dimension, or Project plus dimension. If TRUE, this template only applies to GLFSTMPLT1\_DIV. If FALSE, user will be prompted for division at run time.** |
| **Define a constant project for filtering** | **GLFSTMPLT1\_CONSTANTPROJ** | **Checkbox** | **-** | **NULL unless GLFSTMPLT1\_SCOPE = Single project or Project plus dimension. NULL if GLFSTMPLT1\_REPORTTYPE = Balance sheet. If TRUE, this template only applies to GLFSTMPLT1\_PROJ. If FALSE, user will be prompted for project at run time.** |
| **Filter whole report on a single dimension** | **GLFSTMPLT1\_FILTERONDIMNUM** | **Checkbox** | **-** | **NULL unless GLFSTMPLT1\_SCOPE = Company plus dimensions or Single div plus dimensions or Single proj plus dimensions. If TRUE, the whole report will be confined to a single dim #. If FALSE, use of dimensions (if any) is confined to column definitions (in GLFSTMPLT3).** |
| **Filter whole report on a value for a single dimension** | **GLFSTMPLT1\_FILTERONDIMVAL** | **Checkbox** | **-** | **NULL unless GLFSTMPLT1\_FILTERONDIMNUM is TRUE. If TRUE, the whole report will be confined to a single pair of dim # and dim value. If FALSE, the report will be confined to a single dim # but may (depending on how columns are defined in GLFSTMPLT3) include all values for that dim #.** |
| **Define a constant dimension for filtering** | **GLFSTMPLT1\_CONSTANTFILTERDIMNUM** | **Checkbox** | **-** | **NULL unless GLFSTMPLT1\_FILTERONDIMNUM is TRUE. If TRUE, GLFSTMPLT1\_CONSTANTDIMNUM is the # of the constant dimension. If FALSE, user will be prompted for dimension value at run time.** |
| **Define a constant dimension value for filtering** | **GLFSTMPLT1\_CONSTANTFILTERDIMVAL** | **Checkbox** | **-** | **NULL unless GLFSTMPLT1\_FILTERONDIMVAL is TRUE. If TRUE, GLFSTMPLT1\_CONSTANTDIMNUM is the # of the constant dimension and GLFSTMPLT1\_CONSTANTDIMVAL is the value. If FALSE, user will be prompted for dimension value at run time.** |
| **Filter report on this division** | **GLFSTMPLT1\_DIV** | **Lookup to GLDIV** | **-** | **NULL unless GLFSTMPLT1\_CONSTANTDIV is TRUE.** |
| **Filter report on this project** | **GLFSTMPLT1\_PROJ** | **Lookup to GLPROJ** | **-** | **NULL unless GLFSTMPLT1\_CONSTANTPROJ is TRUE.** |
| **Filter report on this dimension** | **GLFSTMPLT1\_CONSTANTDIMNUM** | **Number(2,0)** | **-** | **NULL unless GLFSTMPLT1\_CONSTANTFILTERDIMNUM is TRUE.** |
| **Filter report on this dimension value** | **GLFSTMPLT1\_CONSTANTDIMVAL** | **Text(20)** | **-** | **NULL unless GLFSTMPLT1\_CONSTANTFILTERDIMVAL is TRUE.** |
|  | **EXTERNAL\_ID** |  |  | **GLFSTMPLT1\_CMPNO + GLFSTMPLT1\_TEMPLATE** |

*Table 1 –* ***GLFSTMPLT1***

Table **GLFSTMPLT2** – Financial statement template table # 2: header table for financial statement column definitions.

| **Field label** | **Column** | **Type/size** | **Possible values** | **Meaning / comment** |
| --- | --- | --- | --- | --- |
|  | **GLFSTMPLT2\_GLFSTMPLT1** | **Master/detail to GLFSTMPLT1** | **-** |  |
| **ID of this variant of the template** | **GLFSTMPLT2\_FORMAT** | **Text(30)** | **Not NULL** | **Identifies a variant of the template. Cannot be NULL. If no variants are defined, the GLFSTMPLT2\_FORMAT for the first one is defaulted to "Original version."** |
| **Number of columns on report** | **GLFSTMPLT2\_COLUMNS** | **Number(2,0)** | **> 0** | **Statements are columnar. The leftmost column of every statement (column zero) always contains text and never has a header; this column is assumed. GLFSTMPLT2\_COLUMNS refers to the number of data columns to be printed to the right of the text.** |
| **Number of tab-spaces to left of first column** | **GLFSTMPLT2\_COLZEROOFFSET** | **Number(2,0)** | **>= 0** | **Number of tab-spaces by which column zero is offset from the left edge of the page.** |
| **Width of leftmost (text) column in characters** | **GLFSTMPLT2\_COLZEROWIDTH** | **Number(2,0)** | **> 0** | **Width of column zero in characters. If more than GLFSTMPLT2\_COLZEROWIDTH characters are entered for any line in column zero, the text will wrap.** |
| **Maximum number of lines in any column heading** | **GLFSTMPLT2\_HEADERLINES** | **Number(1,0)** | **> 0 and <= 5** | **Number of column header lines (1, 2, 3, 4 or 5)** |
| **Currency precision** | **GLFSTMPLT2\_CURRPRECISION** | **Number(1,0)** | **0 or 2** | **The decimal precision of the reporting currency. Defaults to 2 but may be overridden to zero (as for Yen).** |
| **Currency rounding** | **GLFSTMPLT2\_ROUNDCURR** | **Picklist** | **Don't round** | **The default.** |
| **Round to even units** | **Round to even units of currency.** |
| **Round to thousands** | **Round to even thousands.** |
| **Round to millions** | **Round to even millions.** |
| **What to round** | **GLFSTMPLT2\_ROUNDWHAT** | **Picklist** | **NULL** | **If GLFSTMPLT2\_ROUNDCURR = Don't round** |
| **Lines only** | **Only round the numbers that appear on report lines. Add the unrounded line value to totals. When the value of the line is referenced in a calculation, use the unrounded value.** |
| **Lines and totals** | **Round the numbers that appear on report lines, and add the rounded values to totals. When the value of the line is referenced in a calculation, use the unrounded value.** |
| **Lines and calculation values** | **Round the numbers that appear on report lines. Add the unrounded line value to totals. When the value of the line is referenced in a calculation, use the rounded value.** |
| **Everything** | **Round the numbers that appear on report lines. Add the rounded line value to totals. When the value of the line is referenced in a calculation, use the rounded value.** |
| **How negative numbers are identified** | **GLFSTMPLT2\_MINUSSIGN** | **Picklist** | **Parentheses** | **The default. (999)** |
| **Leading dash** | **-999** |
| **Trailing dash** | **999-** |
| **Decimal point and thousands separator** | **GLFSTMPLT2\_PUNCTUATION** | **Picklist** | **Decimal = dot, 1000s = comma** | **Use dot for decimal point and comma for thousands separator: 999,999,999.99** |
| **Decimal = comma, 1000s = dot** | **Use comma for decimal point and dot for thousands separator: 999.999.999,99** |
| **Percentage / small number precision** | **GLFSTMPLT2\_PCTPRECISION** | **Number(1,0)** | **0, 1, or 2** | **Number of factional digits to print for percentages and other small numbers.** |
| **Character to print when a number column is empty** | **GLFSTMPLT2\_NULLVALUEINDICATOR** | **Text(1)** | **"-" or <blank>** | **Character (dash or blank) to display at the intersections of calculated lines and calculated columns. Regardless of the value of GLFSTMPLT2\_NULLVALUEINDICATOR, blanks are output when exporting reports.** |
| **Print a dash when a number column equals zero** | **GLFSTMPLT2\_SHOWDASHFORZEROVALUE** | **Checkbox** | **-** | **If GLFSTMPLT2\_SHOWDASHFORZEROVALUE is TRUE, a dash is output in the position immediately to the left of the decimal indicator when number columns equal zero; otherwise a formatted zero is output. Regardless of the value of GLFSTMPLT2\_ZEROVALUEINDICATOR, zeroes are output when exporting reports.** |
| **Number of centered title lines at the top of the page** | **GLFSTMPLT2\_TITLELINES** | **Number(2,0)** | **>=0 and < 11** | **Number of title lines at top of report (title lines are centered)** |
| **Top title line** | **GLFSTMPLT2\_TITLELINE1** | **Text(70)** | **-** | **NULL if GLFSTMPLT2\_TITLELINES = 0.  Predefined macros that can be embedded in GLFSTMPLT3\_TITKELINE1...10 and resolved at run time:**  **"CO" = Company # [GLCNTL\_CMPNO]  "COMPANY" = Company name [GLCNTL\_CMPNO\_\_R.SYCMP\_DESCR]  "PROJ" = Project number [GLPROJ\_PJPROJ]  "PROJECT" = Project title [GLPROJ\_PJPROJ\_\_R.PJPROJ\_DESCR]  "DIV" = Division code [GLDIV\_DIV]  "DIVISION" = Division name [GLDIV\_DESCR]  "MMMM" = January, February, etc., as of report date  "MMM" = Jan, Feb, etc., as of report date  "MM" = month of report as 01, 02, etc.  "YYYY" = year of report date as 2014, 2015, etc.  "YY" = year of report date as 14, 15, etc.  "DD" = day of report date as 1,2…30,31  "DIM#" = dimension # (1-8)  "DIMSRC" = dimension source name  "DIMVAL" = dimension value  "[nn]" (where nn is a number from 1 to 70) can be appended to any token (e.g. DIVISION[30]). If the value of token exceeds nn characters in length, only the leftmost nn characters are used.** |
| **Second title line** | **GLFSTMPLT2\_TITLELINE2** | **Text(70)** | **-** | **NULL if GLFSTMPLT2\_TITLELINES < 2** |
| **Third title line** | **GLFSTMPLT2\_TITLELINE3** | **Text(70)** | **-** | **NULL if GLFSTMPLT2\_TITLELINES < 3** |
| **Fourth title line** | **GLFSTMPLT2\_TITLELINE4** | **Text(70)** | **-** | **NULL if GLFSTMPLT2\_TITLELINES < 4** |
| **Fifth title line** | **GLFSTMPLT2\_TITLELINE5** | **Text(70)** | **-** | **NULL if GLFSTMPLT2\_TITLELINES < 5** |
| **Sixth title line** | **GLFSTMPLT2\_TITLELINE6** | **Text(70)** | **-** | **NULL if GLFSTMPLT2\_TITLELINES < 6** |
| **Seventh title line** | **GLFSTMPLT2\_TITLELINE7** | **Text(70)** | **-** | **NULL if GLFSTMPLT2\_TITLELINES < 7** |
| **Eighth title line** | **GLFSTMPLT2\_TITLELINE8** | **Text(70)** | **-** | **NULL if GLFSTMPLT2\_TITLELINES < 8** |
| **Ninth title line** | **GLFSTMPLT2\_TITLELINE9** | **Text(70)** | **-** | **NULL if GLFSTMPLT2\_TITLELINES < 9** |
| **Tenth title line** | **GLFSTMPLT2\_TITLELINE10** | **Text(70)** | **-** | **NULL if GLFSTMPLT2\_TITLELINES < 10** |
| **~~Standard # of column widths to indent first column~~** | **~~GLFSTMPLT2\_COL1STDINDENT~~** | **~~Number(1,0)~~** | **~~>= 0 <= 4~~** | **~~Number of column widths to indent Column 1 from its normal starting position on the page. Defaults to zero and can only be overridden if GLFSTMPLT2\_COLUMNS = 1. When overridden, Column 1 for a given line may be moved to the left by setting GLFSTMPLT4\_NEGATIVECOL1INDENT to some value <= GLFSTMPLT2\_COL1STDINDENT. The purpose of GLFSTMPLT2\_COL1STDINDENT and GLFSTMPLT4\_NEGATIVECOL1INDENT is to print "old style" balance sheets & schedules in a single, staggered column. Both GLFSTMPLT2\_COL1STDINDENT and GLFSTMPLT4\_NEGATIVECOL1INDENT are ignored when exporting a report to spreadsheet software.~~** |
| **Unresolved line labels exist** | **GLFSTMPLT2\_UNRESOLVEDLABELS** | **Checkbox** | **-** | **No user input - set by trigger when format is saved.  If GLFSTMPLT2\_UNRESOLVEDLABELS is TRUE, no report can be crated from the format. If there are unresolved line labels - e.g. GLFSTMPLT3\_VALUE*n*CALCLABEL is not NULL but no corresponding GLFSTMPLT4\_CALCLABEL exists - the template is unusable, the user is warned, and GLFSTMPLT2\_UNRESOLVEDLABELS is set to TRUE - meaning that statement calculation cannot be run for GLFSTMPLT2\_FORMAT.** |
| **Looping calculations are defined** | **GLFSTMPLT2\_LOOPINCALCULATION** | **Checkbox** | **-** | **Not maintainable by user. Initialized to NULL. When any GLFSTMPLT3, GLFSTMPLT4, or GLFSTMPLT5 is saved:  GLFSTMPLT2\_LOOPINCALCULATION is set to FALSE if no looping calculations are defined.  GLFSTMPLT2\_LOOPINCALCULATION is set to TRUE if looping calculations exist. If GLFSTMPLT2\_LOOPINCALCULATION is TRUE when a user attempts to generate a report, the report generation aborts with an error message (e.g. "Cannot create statement - loop in calculations exists.")** |
|  |  |  |  | **The following columns permit overrides to GLFSTMPLT1 data. They are NULL unless GLFSTMPLT1-level data is being overwritten.** |
| **Override template-level report type** | **GLFSTMPLT2\_OVERRIDETYPE** | **Checkbox** | **-** | **If TRUE, override GLFSTMPLT1\_REPORTTYPE** |
| **Override template-level report scope** | **GLFSTMPLT2\_OVERRIDESCOPE** | **Checkbox** | **-** | **If TRUE, override GLFSTMPLT1\_SCOPE** |
| **Override template-level division filter** | **GLFSTMPLT2\_OVERRIDEDIVFILTER** | **Checkbox** | **-** | **If TRUE, override GLFSTMPLT1 division filters** |
| **Override template-level project filter** | **GLFSTMPLT2\_OVERRIDEPROJFILTER** | **Checkbox** | **-** | **If TRUE, override GLFSTMPLT1 project filters** |
| **Override template-level dimension filter** | **GLFSTMPLT2\_OVERRIDEDIMFILTER** | **Checkbox** | **-** | **If TRUE, override GLFSTMPLT1 dimension filters** |
| **Report type** | **GLFSTMPLT2\_REPORTTYPE** | **Picklist** | **NULL** | **If GLFSTMPLT2\_OVERRIDETYPE = FALSE** |
| **Balance sheet** | **Formal balance sheet** |
| **Balance sheet schedule** | **Supporting schedule to balance sheet** |
| **P & L** | **Formal P & L** |
| **P & L schedule** | **Supporting schedule to P & L** |
| **Cash flow statement** | **Cash flow statement (aka statement of changes to financial position). Not supported in initial release.** |
| **Cash flow stmt schedule** | **Supporting schedule to cash flow statement. Not supported in initial release.** |
| **Report scope** | **GLFSTMPLT2\_SCOPE** | **Picklist** | **NULL Company wide Single division Single project Company plus dimension Division plus dimension Project plus dimension** | **NULL if GLFSTMPLT2\_OVERRIDESCOPE = FALSE Must be Company wide if GLFSTMPLT2\_REPORTTYPE = Balance sheet. Must be Company wide or Company plus dimension if GLFSTMPLT2\_REPORTTYPE = Balance sheet schedule.** |
| **Consolidated or Single-Company Report?** | **GLFSTMPLT2\_ONECOMPORCONSOL** | **Picklist** | **Single Company**  **Consolidated** | **Default to Single Company.**  **If Single Company, GLFSTMPLT5\_CMPNO must be Reporting Company.**  **If Consolidated, GLFSTMPLT5\_CMPNO may be any company in the chain of consolidation.** |
| **Define an constant division for filtering** | **GLFSTMPLT2\_CONSTANTDIV** | **Checkbox** | **-** | **NULL unless GLFSTMPLT2\_SCOPE = Single division, Single project, Division plus dimension, or Project plus dimension. If TRUE, this template only applies to GLFSTMPLT2\_DIV. If FALSE, user will be prompted for division at run time.** |
| **Define an constant project for filtering** | **GLFSTMPLT2\_CONSTANTPROJ** | **Checkbox** | **-** | **NULL unless GLFSTMPLT2\_SCOPE = Single project or Project plus dimension. NULL if GLFSTMPLT2\_REPORTTYPE = Balance sheet. If TRUE, this template only applies to GLFSTMPLT2\_PROJ. If FALSE, user will be prompted for project at run time.** |
| **Filter report on this division** | **GLFSTMPLT2\_DIV** | **Lookup to GLDIV** | **-** | **NULL unless GLFSTMPLT2\_CONSTANTDIV is TRUE.** |
| **Filter report on this project** | **GLFSTMPLT2\_PROJ** | **Lookup to GLPROJ** | **-** | **NULL unless GLFSTMPLT2\_CONSTANTPROJ is TRUE.** |
| **Filter whole report on a single dimension** | **GLFSTMPLT2\_FILTERONDIMNUM** | **Checkbox** | **-** | **NULL unless GLFSTMPLT2\_SCOPE = Company plus dimensions or Single div plus dimensions or Single proj plus dimensions. If TRUE, the whole report will be confined to a single dim #. If FALSE, use of dimensions (if any) is confined to column definitions (in GLFSTMPLT3).** |
| **Filter whole report on a value for a single dimension** | **GLFSTMPLT2\_FILTERONDIMVAL** | **Checkbox** | **-** | **NULL unless GLFSTMPLT2\_FILTERONDIMNUM is TRUE. If TRUE, the whole report will be confined to a single pair of dim # and dim value. If FALSE, the report will be confined to a single dim # but may (depending on how columns are defined in GLFSTMPLT3) include all values for that dim #.** |
| **Define an constant dimension for filtering** | **GLFSTMPLT2\_CONSTANTFILTERDIMNUM** | **Checkbox** | **-** | **NULL unless GLFSTMPLT2\_FILTERONDIMNUM is TRUE. If TRUE, GLFSTMPLT2\_CONSTANTFILTERDIMNUM is the # of the constant dimension. If FALSE, user will be prompted for dimension value at run time.** |
| **Define a constant dimension value for filtering** | **GLFSTMPLT2\_CONSTANTFILTERDIMVAL** | **Checkbox** | **-** | **NULL unless GLFSTMPLT2\_FILTERONDIMVAL is TRUE. If TRUE, GLFSTMPLT2\_CONSTANTFILTERDIMNUM is the # of the constant dimension and GLFSTMPLT2\_CONSTANTFILTERDIMVAL is the value. If FALSE, user will be prompted for dimension value at run time.** |
| **Filter report on this dimension** | **GLFSTMPLT2\_CONSTANTDIMNUM** | **Number(2,0)** | **-** | **NULL unless GLFSTMPLT2\_CONSTANTFILTERDIMNUM is TRUE.** |
| **Filter report on this dimension value** | **GLFSTMPLT2\_CONSTANTDIMVAL** | **Text(20)** | **-** | **NULL unless GLFSTMPLT2\_CONSTANTFILTERDIMVAL is TRUE.** |
|  |  |  |  | **The following columns are set in Process 9** |
| **Last report number created** | **GLFSTMPLT2\_LASTREPORT** | **Loolup to GLFINSTMT1** | **-** | **The most recent report created from this template.** |
| **The template is in error** | **GLFSTMPLT2\_INERROR** | **Checkbox** | **-** | **The statement cannot be created.** |
| **There is a loop in the calculations** | **GLFSTMPLT2\_LOOPINGCALCULATION** | **Checkbox** | **-** | **There is a loop in the calculation – a calculation depends on its own value.** |
| **There is an infinite regression** | **GLFSTMPLT2\_INFINITEREGRESS** | **Checkbox** | **-** | **A total includes a calculation line that references a total that includes a calculation line that references a total – and so on, infinitely.** |
| **The template is incomplete** | **GLFSTMPLT2\_INCOMPLETETEMPLATE** | **Checkbox** | **-** | **Lines are referenced that don’t exist ot columns are not defined.** |
|  | **EXTERNAL\_ID** |  |  | **GLFSTMPLT2\_GLFSTMPLT1 + GLFSTMPLT2\_FORMAT** |

*Table 2 –* ***GLFSTMPLT2***

Table **GLFSTMPLT3** – Financial statement template table # 3: financial statement column definitions.

| **Field label** | **Column** | **Type/size** | **Possible values** | **Meaning / comment** |
| --- | --- | --- | --- | --- |
|  | **GLFSTMPLT3\_GLFSTMPLT2** | **Master/detail to GLFSTMPLT2** | **-** |  |
| **Column number** | **GLFSTMPLT3\_COLNO** | **Number(2,0)** | **> 0 and <= GLFSTMPLT2\_COLUMNS** | **Column offset from column zero.** |
| **Non-printing column flag** | **GLFSTMPLT3\_NONPRINTING** | **Checkbox** | **-** | **If TRUE, this column is not printed. Its purpose is to be used to calculate the value of calculated columns (GLFSTMPLT3\_VALUETYPE = Calculated).** |
| **Print column values as percentages or other small numbers** | **GLFSTMPLT3\_PRINTASPERCENTAGE** | **Checkbox** | **-** | **Defaults to FALSE. If set to TRUE, print every line in column as a percentage. A TRUE GLFSTMPLT4\_PRINTASPERCENTAGE has precedence over a FALSE GLFSTMPLT3\_PRINTASPERCENTAGE. A TRUE GLFSTMPLT3\_PRINTASPERCENTAGE has precedence over a FALSE GLFSTMPLT4\_PRINTASPERCENTAGE.** |
| **Top line of column heading** | **GLFSTMPLT3\_HEADERLINE1** | **Text(30)** | **-** | **NULL if GLFSTMPLT3\_NONPRINTING = TRUE; otherwise, this is the top header line.   Predefined macros that can be embedded in GLFSTMPLT3\_HEADERLINE1,2 & 3 and resolved at run time:**  **"MMMM" = January, February, etc., as of column period start date  "MMM" = Jan, Feb, etc., as of column period start date  "MM" = month of column period start date as 01, 02, etc.  "NNNN" = January, February, etc., as of column period end date  "NNN" = Jan, Feb, etc., as of column period end date  "NN" = month of column period end date as 01, 02, etc.  "PP" = column period # as 1,2…13  "YYY" = year of column period start date as 2014, 2015, etc.  "YY" = year of column period start date as 14, 15, etc.  "ZZZ" = year of column period end date as 2014, 2015, etc.  "ZZ" = year of column period end date as 14, 15, etc.  "SS" = day of column period start date as 1,2…30,31  "EE" = day of column period end date as 1,2…30,31  "DIV" = division ID  "DIVISION" = division name  "PROJ" = Project ID  "DIM#" = dimension # (1-8)  "DIMSRC" = dimension source name  "DIMVAL" = dimension value  "[nn]" (where nn is a number from 1 to 30) can be appended to any token (e.g. DIVISION[30]). If the value of token exceeds nn characters in length, only the leftmost nn characters are used.** |
| **Second line of column heading** | **GLFSTMPLT3\_HEADERLINE2** | **Text(30)** | **-** | **Defined if GLFSTMPLT3\_NONPRINTING = FALSE and GLFSTMPLT2\_HEADERLINES > 1; otherwise NULL.** |
| **Third line of column heading** | **GLFSTMPLT3\_HEADERLINE3** | **Text(30)** | **-** | **Defined if GLFSTMPLT3\_NONPRINTING = FALSE and GLFSTMPLT2\_HEADERLINES > 2; otherwise NULL.** |
| **Fourth line of column heading** | **GLFSTMPLT3\_HEADERLINE4** | **Text(30)** | **-** | **Defined if GLFSTMPLT3\_NONPRINTING = FALSE and GLFSTMPLT2\_HEADERLINES > 3; otherwise NULL.** |
| **Fifth line of column heading** | **GLFSTMPLT3\_HEADERLINE5** | **Text(30)** | **-** | **Defined if GLFSTMPLT3\_NONPRINTING = FALSE and GLFSTMPLT2\_HEADERLINES > 4; otherwise NULL.** |
| **Type and origin of column values** | **GLFSTMPLT3\_VALUETYPE** | **Picklist** | **GL** | **If GL,   read GLAYSTMT where GLAYSTMT\_SOURCE = GL data.** |
| **Statement entry** | **If Statement entry,   read GLAYSTMT where GLAYSTMT\_SOURCE = Statement entry data.** |
| **GL + Statement entry** | **If GL + Statement entry,   read GLAYSTMT where GLAYSTMT\_SOURCE = GL data or Statement entry data.** |
| **Budget** | **If Budget and GLFSTMPLT3\_BUDSTATUS = Official base budget,   read GLAYSTMT where GLAYSTMT\_SOURCE = Budget data and GLACCTBUD\_REVSTATUS for GLAYSTMT\_BUDGETREVISION = Official base budget.   If Budget and GL FSTMPLT3\_BUDSTATUS = Official revision,   read GLAYSTMT where GLAYSTMT\_SOURCE = Budget data and GLACCTBUD\_REVSTATUS for GLAYSTMT\_BUDGETREVISION = Official revision and GLACCTBUD\_REVDATE is the latest GLACCTBUD\_REVDATE.** |
| **Calculated** | **If Calculated, the formula is defined below and in GLDFTMPLT4 as follows:   Every formula has the form VALUE1 OPERATOR VALUE2.   There are six OPERATORs available for formulas: Plus, Minus, Multiplied by, Divided by, As a percentage of, and No operator. When No operator is selected, VALUE1 is the value of the calculation and VALUE2 is NULL.   VALUE1 and/or VALUE2 may be constants; if so, they stored here as GLFSTMPLT3\_VALUE1CONSTANT and/or GLFSTMPLT3\_VALUE2CONSTANT.    Otherwise, VALUEx is defined in two parts:   (1) A COLUMN component, which is defined here as GLFSTMPLT3\_VALUExCOL and which equals a valid GLFSTMPLT3\_COLNO. The GLFSTMPLT3\_COLNO selected must be < than the current record's GLFSTMPLT3\_COLNO (to prevent looping formulas).   (2) A ROW component, which is defined in GLFSTMPLT4 as GLFSTMPLT4\_VALUExROW.** |
| **Status for identifying budget data** | **GLFSTMPLT3\_BUDSTATUS** | **Picklist** | **Official base budget Official revision** | **NULL if GLFSTMPLT3\_VALUETYPE not = Budget Data. Otherwise, determines the flavor of budget data to be retrieved from GLAYSTMT.** |
| **How revision level for identifying budget data is known** | **GLFSTMPLT3\_BUDREVLEVELMETHOD** | **Picklist** | **Constant Variable** | **NULL unless GLFSTMPLT3\_BUDSTATUS = Official revision. Otherwise, the method by which GLACCTBUD\_REVLEVEL will be identified.** |
| **Constant revision level** for identifying budget data | **GLFSTMPLT3\_BUDREVLEVEL** | **Text(10)** | **-** | **NULL unless GLFSTMPLT3\_BUDREVLEVELMETHOD = Constant. Otherwise, the GLACCTBUD\_REVLEVEL to select from.** |
| **How effective year of column is known** | **GLFSTMPLT3\_CURRENTYEARMEANS** | **Picklist** | **Floating Absolute Entered at run time**  **Calculated column**  **Floating period with offset**  **Floating year with offset** | **This defines the meaning of "current YEAR":  If GLFSTMPLT3\_CURRENTYEARMEANS = Floating, then the "current YEAR" is the YEAR of the reporting period.  If GLFSTMPLT3\_CURRENTYEARMEANS = Absolute, then the "current YEAR" = GLFSTMPLT3\_FIXEDYEARNO regardless of what YEAR is actually being closed.  If GLFSTMPLT3\_CURRENTYEARMEANS = Entered at run time, then the user is prompted for "current YEAR" when the actual report is calculated.**  **If GLFSTMPLT3\_VALUETYPE = Calculated, then GLFSTMPLT3\_CURRENTYEARMEANS is forced to Calculated column. No user override is allowed. Calculated column cannot be selected as a value for GLFSTMPLT3\_CURRENTYEARMEANS by the user.**  **If GLFSTMPLT3\_CURRENTPERIODMEANS = Floating period with offset, then GLFSTMPLT3\_CURRENTYEARMEANS must also equal Floating period with offset – meaning that the year is determined by adding GLFSTMPLT3\_PERIODOFFSET to the reporting period.**  **If GLFSTMPLT3\_CURRENTPERIODMEANS = Floating year with offset, then the “current YEAR” is the YEAR of the reporting period plus GLFSTMPLT3\_YEAROFFSET.** |
| **How effective period of column is known** | **GLFSTMPLT3\_CURRENTPERIODMEANS** | **Picklist** | **Floating Absolute Entered at run time Calculated column**  **Floating period with offset** | **This defines the meaning of "current period":  If GLFSTMPLT3\_CURRENTPERIODMEANS = Floating, then the "current period" is the reporting period.  If GLFSTMPLT3\_CURRENTPERIODMEANS = Absolute, then the "current period" = GLFSTMPLT3\_FIXEDPERIODNO regardless of what period is actually being closed.  If GLFSTMPLT3\_CURRENTPERIODMEANS = Entered at run time, then the user is prompted for "current period" when the actual report is calculated.  If GLFSTMPLT3\_VALUETYPE = Calculated, then GLFSTMPLT3\_CURRENTPERIODMEANS is forced to Calculated column. No user override is allowed. Calculated column cannot be selected as a value for GLFSTMPLT3\_CURRENTPERIODMEANS by the user.**  **If GLFSTMPLT3\_CURRENTPERIODMEANS = Floating period with offset, then the “current period” is the reporting period plus GLFSTMPLT3\_PERIODOFFSET.** |
| **Constant effective year of column** | **GLFSTMPLT3\_FIXEDYEAR** | **Number(4,0)** | **-** | **NULL if GLFSTMPLT3\_CURRENTYEARMEANS != Absolute. Else, GLFSTMPLT3\_FIXEDYEAR is the year of the column.** |
| **Constant effective period of column** | **GLFSTMPLT3\_FIXEDPERIODNO** | **Number(2,0)** | **NULL or ( > 0 and < 14)** | **NULL if GLFSTMPLT3\_CURRENTPERIODMEANS != Absolute. Else, GLFSTMPLT3\_FIXEDPERIODNO is the number of the period for the column.** |
| **Offset from reporting period** | **GLFSTMPLT3\_PERIODOFFSET** | **Number(2,0)** | **> -100 and < 100** | **NULL if GLFSTMPLT3\_CURRENTPERIODMEANS != Floating period with offset,**  **Else, number of periods by which the “current period” is offset from the reporting period.** |
| **Offset from reporting year** | **GLFSTMPLT3\_YEAROFFSET** | **Number(1,0)** | **> -10 and < 10** | **NULL if GLFSTMPLT3\_CURRENTYEARMEANS != Floating year with offset,**  **Else, number of years by which the “current year” is offset from the year of the reporting period.** |
| **How to treat period zero if period is offset** | **GLFSTMPLT3\_PERIODZEROPOLICY** | **Picklist** | **Skip**  **Add to period 1**  **Show by itself** | **This defines how to treat period zero if GLFSTMPLT3\_CURRENTPERIODMEANS = Floating period with offset:**  **If GLFSTMPLT3\_PERIODZEROPOLICY = Skip, skip period zero.**  **If GLFSTMPLT3\_PERIODZEROPOLICY = Add to period 1, add the period zero balance to the balance for period 1 and don’t show period zero separately.**  **If GLFSTMPLT3\_PERIODZEROPOLICY = Show by itself, show period zero in a column of its own.** |
| **How to treat period 13 if period is offset** | **GLFSTMPLT3\_PERIOD13POLICY** | **Picklist** | **Skip**  **Add to period 12**  **Show by itself** | **This defines how to treat period 13 if GLFSTMPLT3\_CURRENTPERIODMEANS = Floating period with offset:**  **If GLFSTMPLT3\_PERIOD13POLICY = Skip, skip period 13.**  **If GLFSTMPLT3\_PERIOD13POLICY = Add to period 12, add the period 13 balance to the balance for period 12 and don’t show period 13 separately.**  **If GLFSTMPLT3\_PERIOD13POLICY = Show by itself, show period 13 in a column of its own.** |
| **Timeframe of column** | **GLFSTMPLT3\_PTDYTD** | **Picklist** | **Period totals Year to date totals YTD period average Calculated column** | **In the following, "current period" is defined by GLFSTMPLT3\_CURRENTPERIODMEANS, as possibly modified by GLFSTMPLT3\_COLUMNSTARTOFFSET, below.  If GLFSTMPLT3\_PTDYTD = Year to date, then the column value is the sum of periods zero (the beginning balance) through the "current period." If GLFSTMPLT3\_PTDYTD = Period totals, then the column value is the value for the "current period." If GLFSTMPLT3\_PTDYTD = YTD period average, then the column value is the average of the periods zero through the "current period." If GLFSTMPLT3\_VALUETYPE = Calculated, then GLFSTMPLT3\_PTDYTD is forced to Calculated column. No user override is allowed. Calculated column cannot be selected as a value for GLFSTMPLT3\_PTDYTD by the user.** |
| **Label of external data exported from column** | **GLFSTMPLT3\_EXPORTLABEL** | **Text(50)** | **-** | **NULL by default. It is pointless for more than one column to share a GLFSTMPLT3\_EXPORTLABEL, since the results of the higher-numbered column will simply overwrite the results of the lower-numbered column. If, for any intersection of line and column, GLFSTMPLT4\_EXPORTLABEL*n* = GLFSTMPLT3\_EXPORTLABEL, the value of the intersection is written to GLFSTMPLT6 at report generation with:  GLFSTMPLT6\_CMPNO = GLFSTMPLT1\_CMPNO  GLFSTMPLT6\_LABEL = GLFSTMPLT3\_EXPORTLABEL  GLFSTMPLT6\_FROMYEAR = Column year  GLFSTMPLT6\_FROMPERIOD = Column period  GLFSTMPLT6\_THRUYEAR = Column year  GLFSTMPLT6\_THRUPERIOD = Column period  GLFSTMPLT6\_SCALE = Currency  GLFSTMPLT6\_CURRENCYVAL = line/column value  GLFSTMPLT6\_GENERALVAL = NULL** |
|  |  |  |  | **The following fields define filters that apply to the whole report column.  NOTE: it is perfectly possible to define a column whose potential contents are entirely filtered out by the report-level filters defined in GLFSTMPLT1. In the case of conflicting report-level (GLFSTMPLT1) and column-level (GLFSTMPLT3) definitions, the report level takes precedence.  It is also perfectly possible for the line filters defined in GLFSTMPLT4 to conflict with the column filters defined in GLFSTMPLT3. In that case, the intersection of the conflicting column and line will be null.** |
| **Filter on division** | **GLFSTMPLT3\_FILTERONDIV** | **Checkbox** |  | **Default value is FALSE. If set to TRUE, GLFSTMPLT3\_FILTERONPROJ can also be set to TRUE.** |
| **Filter on project** | **GLFSTMPLT3\_FILTERONPROJ** | **Checkbox** |  | **Default value is FALSE. If GLFSTMPLT3\_FILTERONDIV is set to TRUE, GLFSTMPLT3\_FILTERONPROJ can also be set to TRUE. If GLFSTMPLT3\_FILTERONDIV is set to FALSE, GLFSTMPLT3\_FILTERONPROJ must also be set to FALSE (since projects are assigned to divisions by definition).** |
| **Filter on dimension** | **GLFSTMPLT3\_FILTERONDIM** | **Checkbox** |  | **Default value is FALSE. Can be set to TRUE regardless of the values of GLFSTMPLT3\_FILTERONDIV and GLFSTMPLT3\_FILTERONPROJ.** |
| **Division filtering method** | **GLFSTMPLT3\_DIVMETHOD** | **Picklist** | **NULL** | **If GLFSTMPLT3\_FILTERONDIV is FALSE.** |
| **All divisions** | **Overrides any report-level division filters // 11/18/16** |
| **Variable division** | **User will enter division for this column at runtime.** |
| **Constant division** | **Division for this column will be defined in GLFSTMPLT3\_DIV.** |
| **No division** | **This value cannot be selected if GLFSTMPLT3\_FILTERONPROJ is TRUE because projects are division-specific.  Column values retrieved from GLAYSTMT for an account with have a division-specific GLAYSTMT\_GLSOURCE (corresponding to GLAYPERPROJ, GLAYPERDIV, GLAYPERDIMPROJ, or GLAYPERDIMDIV), where division is NULL. This will return only values where no division was entered on the originating transaction (in GLTXN). This is not at all the same as setting GLFSTMPLT3\_FILTERONDIV to FALSE - that will return account-wide values, regardless of whether a division was entered on the origination transaction or not.** |
| **Constant division for filtering** | **GLFSTMPLT3\_DIV** | **Lookup to GLDIV** |  | **NULL unless GLFSTMPLT3\_DIVMETHOD is Constant division. Otherwise, this is the division for this column.** |
| **Project filtering method** | **GLFSTMPLT3\_PROJMETHOD** | **Picklist** | **NULL** | **If GLFSTMPLT3\_FILTERONPROJ is FALSE.** |
| **All projects** | **Overrides any report-level project filters // 11/18/16** |
| **Variable project** | **User will enter project for this column at runtime.** |
| **Constant project** | **Project for this column will be defined in GLFSTMPLT3\_PROJ.** |
| **No project** | **Column values retrieved from GLAYSTMT for an account with have a project-specific GLAYSTMT\_GLSOURCE (corresponding to GLAYPERPROJ or GLAYPERDIMPROJ), where project is NULL. This will return only values where no project was entered on the originating transaction (in GLTXN). This is not at all the same as setting GLFSTMPLT3\_FILTERONPROJ to FALSE - that will return account-wide values, regardless of whether a project was entered on the origination transaction or not.** |
| **Constant project for filtering** | **GLFSTMPLT3\_PROJ** | **Lookup to GLPROJ** |  | **NULL unless GLFSTMPLT3\_CONSTANTPROJ is TRUE. Otherwise, this is the project for this column.** |
| **Dimension filtering method** | **GLFSTMPLT3\_DIMMETHOD** | **Picklist** | **NULL** | **If GLFSTMPLT3\_FILTERONDIM is FALSE.** |
| **All dimensions** | **Overrides any report-level dimension filters // 11/18/16** |
| **Constant dimension, constant value** | **Dimension # is always GLFSTMPLT3\_DIMNUM and dimension value is always GLFSTMPLT3\_DIMVAL.** |
| **Constant dimension, exclude constant value** | **Dimension # is always GLFSTMPLT3\_DIMNUM and dimension value is always GLFSTMPLT3\_DIMVAL.  This setting returns the value of all postings that reference GLFSTMPLT3\_DIMNUM except those that specify GLFSTMPLT3\_DIMVAL.** |
| **Constant dimension, variable value** | **Dimension # is always GLFSTMPLT3\_DIMNUM and dimension value is entered at runtime.** |
| **Constant dimension, exclude variable value** | **Dimension # is always GLFSTMPLT3\_DIMNUM and dimension value is entered at runtime.  This setting returns the value of all postings that reference GLFSTMPLT3\_DIMNUM except those that specify the dimension value entered at runtime.** |
| **Constant dimension, all values** | **Dimension # is always GLFSTMPLT3\_DIMNUM. This setting returns the sum of all postings where GLFSTMPLT3\_DIMNUM is specified.** |
| **Variable dimension, variable value** | **Dimension # and dimension value are entered at runtime.** |
| **Variable dimension, exclude variable value** | **Dimension # and dimension value are entered at runtime.  This setting returns the value of all postings that reference the dimension number entered at runtime except those that specify the dimension value entered at runtime.** |
| **Variable dimension, all values** | **Dimension # is entered at runtime. This setting returns the sum of all postings where the entered dimension # is specified.** |
| **All dimensions, all values** | **This setting returns the sum of all postings where any dimension # is specified.** |
| **No dimension** | **Values retrieved from GLAYSTMT for an account with have a dimension-specific GLAYSTMT\_GLSOURCE (corresponding to GLAYPERDIM, GLAYPERDIMDIV, or GLAYPERDIMPROJ, depending on the values of GLFSTMPLT3\_FILTERONDIM and GLFSTMPLT3\_FILTERONPROJ), where dimension is NULL. This will return only values where no dimension was entered on the originating transaction (in GLTXN).   This is not at all the same as setting GLFSTMPLT3\_FILTERONDIM to FALSE - that will return account-wide or account+division-wide values, regardless of whether a dimension was entered on the origination transaction or not.** |
| **Constant dimension for filtering** | **GLFSTMPLT3\_DIMNUM** | **Number(2,0)** | **Dimension #** | **NULL unless GLFSTMPLT3\_FILTERONDIM is TRUE and GLFSTMPLT3\_DIMMETHOD calls for a constant dimension #.** |
| **Constant dimension value for filtering** | **GLFSTMPLT3\_DIMVAL** | **Text(20)** | **Dimension value** | **NULL unless GLFSTMPLT3\_FILTERONDIM is TRUE and GLFSTMPLT3\_DIMMETHOD calls for a constant dimension value.** |
|  |  |  |  | **The following columns are NULL unless GLFSTMPLT3\_VALUETYPE = Calculated:** |
| **Expression operator for calculation** | **GLFSTMPLT3\_OPERATOR** | **Picklist** | **Plus Minus Multiplied by Divided by As a percentage of No operator** | **If GLFSTMPLT3\_OPERATOR = Plus,   then the value of the column = VALUE1 + VALUE2. If GLFSTMPLT3\_OPERATOR = Minus,   then the value of the column = VALUE1 - VALUE2. If GLFSTMPLT3\_OPERATOR = Multiplied by,   then the value of the column = VALUE1 \* VALUE2. If GLFSTMPLT3\_OPERATOR = Divided by,   then the value of the column = VALUE1 / VALUE2. If GLFSTMPLT3\_OPERATOR = As a percentage of,   then the value of the column = (VALUE1 / VALUE2) \* 100. If GLFSTMPLT3\_OPERATOR = No operator,  then the value of the column = VALUE1.** |
| **Origin of equation left side** | **GLFSTMPLT3\_VALUE1TYPE** | **Picklist** | **Same row, other column Other row, other column Constant Entered at runtime Imported at runtime Special** | **If Same row, other column  VALUE1 for each row will be set to the sum of GLFSTMPLT3\_VALUE1COLFROM through  GLFSTMPLT3\_VALUE1COLTHRU in the same row. (To use one column only, set  GLFSTMPLT3\_VALUE1COLFROM = GLFSTMPLT3\_VALUE1COLTHRU.) If Other row, other column  VALUE1 for each row will be set to the sum of GLFSTMPLT3\_VALUE1COLFROM through   GLFSTMPLT3\_VALUE1COLTHRU in the row whose GLFSTMPLT4\_CALCLABEL =   GLFSTMPLT3\_VALUE1CALCLABEL. If Constant,  VALUE1 for all rows will be set to GLFSTMPLT3\_VALUE1CONSTANT. If Entered at runtime,  VALUE1 for all rows will be entered at runtime.  If Imported at runtime,  VALUE1 for all rows will be read from GLFSTMPLT6 where   GLFSTMPLT6\_CMPNO = GLFSTMPL1\_CMPNO,   GLFSTMPLT6\_YEAR = current year,   GLFSTMPLT6\_PERIOD = current period,   GLFSTMPLT6\_GLDIV = division of column scope (if any),  GLFSTMPLT6\_GLPROJ = project of column scope (if any),  GLFSTMPLT6\_DIMNUM = dimension # of column scope (if any),  GLFSTMPLT6\_DIMVAL = dimension value of column scope (if any), and  GLFSTMPLT6\_LABEL = GLFSTMPLT3\_VALUE1CALCLABEL. If Special,  VALUE1 is determined by GLFSTMPLT3\_VALUE1SPECIALTYPE.  If Same row, other column, VALUE1 will vary from row to row. If Constant or Entered at runtime or Imported at runtime or Special or Other row, other column, VALUE1 is the same throughout the report.** |
| **Origin of equation right side** | **GLFSTMPLT3\_VALUE2TYPE** | **Picklist** | **Same row, other column Other row, other column Constant Entered at runtime Imported at runtime Special** | **NULL if GLFSTMPLT3\_OPERATOR is No operator. Otherwise, If Same row, other column  VALUE2 for each row will be set to the sum of GLFSTMPLT3\_VALUE2COLFROM through  GLFSTMPLT3\_VALUE2COLTHRU in the same row. (To use one column only, set  GLFSTMPLT3\_VALUE2COLFROM = GLFSTMPLT3\_VALUE2COLTHRU.) If Other row, other column  VALUE2 for each row will be set to the sum of GLFSTMPLT3\_VALUE2COLFROM through   GLFSTMPLT3\_VALUE2COLTHRU in the row whose GLFSTMPLT4\_CALCLABEL =   GLFSTMPLT3\_VALUE2CALCLABEL. If Constant,  VALUE2 for all rows will be set to GLFSTMPLT3\_VALUE2CONSTANT. If Entered at runtime,  VALUE2 for all rows will be entered at runtime.  If Imported at runtime,  VALUE2 for all rows will be read from GLFSTMPLT6 where   GLFSTMPLT6\_CMPNO = GLFSTMPL1\_CMPNO,   GLFSTMPLT6\_YEAR = current year,   GLFSTMPLT6\_PERIOD = current period,   GLFSTMPLT6\_GLDIV = division of column scope (if any),  GLFSTMPLT6\_GLPROJ = project of column scope (if any),  GLFSTMPLT6\_DIMNUM = dimension # of column scope (if any),  GLFSTMPLT6\_DIMVAL = dimension value of column scope (if any), and  GLFSTMPLT6\_LABEL = GLFSTMPLT3\_VALUE2CALCLABEL. If Special,  VALUE2 is determined by GLFSTMPLT3\_VALUE2SPECIALTYPE.  If Same row, other column, VALUE2 will vary from row to row. If Constant or Entered at runtime or Imported at runtime or Special or Other row, other column, VALUE2 is the same throughout the report.** |
| **Constant value of equation left side** | **GLFSTMPLT3\_VALUE1CONSTANT** | **Number(16,2)** | **-** | **NULL unless GLFSTMPLT3\_VALUE1TYPE = Constant. Otherwise, this is the value of VALUE1.** |
| **Constant value of equation right side** | **GLFSTMPLT3\_VALUE2CONSTANT** | **Number(16,2)** | **-** | **NULL unless GLFSTMPLT3\_VALUE2TYPE = Constant. Otherwise, this is the value of VALUE2.** |
| **Type of special value for equation left side** | **GLFSTMPLT3\_VALUE1SPECIALTYPE** | **Picklist** | **Period number Periods in year** | **NULL unless GLFSTMPLT3\_VALUE1TYPE = Special. If Period number,  VALUE1 = the current period number. If Periods in year,  VALUE1 = the number of periods in the year.  Examples of use:  To average the period values in the current year,   divide a year to date column by the period number.  To project the year-end balance from a single period,  multiply a column for that period by (periods in the current year - period number + 1)   and add it to the current year to date.** |
| **Type of special value for equation right side** | **GLFSTMPLT3\_VALUE2SPECIALTYPE** | **Picklist** | **Period number Periods in year** | **NULL unless GLFSTMPLT3\_VALUE2TYPE = Special. If Period number,  VALUE2 = the current period number. If Periods in year,  VALUE2 = the number of periods in the year.** |
| **Line label for source of equation left side** | **GLFSTMPLT3\_VALUE1CALCLABEL** | **Text(50)** | **-** | **NULL unless GLFSTMPLT3\_VALUE1TYPE = Other row, other column or Imported at runtime. If GLFSTMPLT3\_VALUE1TYPE = Other row, other column, this label must equal some line label defined as GLFSTMPLT4\_CALCLABEL in GLFSTMPLT4. If GLFSTMPLT3\_VALUE1TYPE = Imported at runtime, this label is used to look up VALUE1 in GLFSTMPLT6.  Typical values of calculation labels, suggesting their usage:  For line labels:  NET\_INCOME  TOTAL\_COST\_OF\_SALES  TOTAL\_G\_S\_AND\_A  For imported data labels:  SQ\_FEET\_IN\_PLANT, SQ\_FEET\_IN\_DEPT  HEADCOUNT\_FOR\_COMPANY, HEADCOUNT\_FOR\_DIVISION  KILOGRAMMES\_SHIPPED\_IN\_PERIOD  WORKING\_DAYS\_IN\_PERIOD  If there are unresolved line labels - e.g. GLFSTMPLT3\_VALUE1CALCLABEL is not NULL but no corresponding GLFSTMPLT4\_CALCLABEL exists - the template is unusable, the user is warned, and GLFSTMPLT2\_UNRESOLVEDLABELS is set to TRUE - meaning that statement calculation cannot be run for GLFSTMPLT2\_FORMAT.** |
| **Line label for source of equation right side** | **GLFSTMPLT3\_VALUE2CALCLABEL** | **Text(50)** | **-** | **NULL unless GLFSTMPLT3\_VALUE2TYPE = Other row, other column or Imported at runtime. If GLFSTMPLT3\_VALUE2TYPE = Other row, other column, this label must equal some line label defined as GLFSTMPLT4\_CALCLABEL in GLFSTMPLT4. If GLFSTMPLT3\_VALUE2TYPE = Imported at runtime, this label is used to look up VALUE2 in GLFSTMPLT6.** |
| **Starting column for source of equation left side** | **GLFSTMPLT3\_VALUE1COLFROM** | **Number(2,0)** | **-** | **NULL if GLFSTMPLT3\_VALUE1TYPE = Constant. Otherwise this is the starting column number in the range of columns in which VALUE1 will be found at runtime. GLFSTMPLT3\_VALUE1COLGFROM must be > 0 and < GLFSTMPLT3\_COLNO (to prevent loops in the calculation).** |
| **Ending column for source of equation left side** | **GLFSTMPLT3\_VALUE1COLTHRU** | **Number(2,0)** | **-** | **NULL if GLFSTMPLT3\_VALUE1TYPE = Constant. Otherwise this is the ending column number in the range of columns in which VALUE1 will be found at runtime. GLFSTMPLT3\_VALUE1COLTHRU must be >= GLFSTMPLT3\_VALUE1COLFROM and < GLFSTMPLT3\_COLNO (to prevent loops in the calculation).** |
| **Starting column for source of equation right side** | **GLFSTMPLT3\_VALUE2COLFROM** | **Number(2,0)** | **-** | **NULL if GLFSTMPLT3\_VALUE2TYPE = Constant or GLFSTMPLT3\_OPERATOR is No operator. Otherwise this is the starting column number in the range of columns in which VALUE2 will be found at runtime. GLFSTMPLT3\_VALUE2COLGFROM must be > 0 and < GLFSTMPLT3\_COLNO (to prevent loops in the calculation).** |
| **Ending column for source of equation right side** | **GLFSTMPLT3\_VALUE2COLTHRU** | **Number(2,0)** | **-** | **NULL if GLFSTMPLT3\_VALUE2TYPE = Constant or GLFSTMPLT3\_OPERATOR is No operator. Otherwise this is the ending column number in the range of columns in which VALUE2 will be found at runtime. GLFSTMPLT3\_VALUE2COLTHRU must be >= GLFSTMPLT3\_VALUE2COLFROM and < GLFSTMPLT3\_COLNO (to prevent loops in the calculation).** |
| **Negate results of expression** | **GLFSTMPLT3\_NEGATEEXPRESSION** | **Checkbox** | **-** | **Defaults to FALSE. If set to TRUE, the sign of the calculation result is flipped (the result is multiplied by -1).  This inverts the default behavior for displaying and totaling calculation results:   When GLFSTMPLT3\_NEGATEEXPRESSION is FALSE and debits are deemed positive (see GLFSTMPLT4\_COMMAND), positive calculation results are deemed debits and negative calculation results are deemed credits.  When GLFSTMPLT3\_NEGATEEXPRESSION is FALSE and credits are deemed positive, positive calculation results are deemed credits and negative calculation results are deemed debits.  When GLFSTMPLT3\_NEGATEEXPRESSION is TRUE and debits are deemed positive, positive calculation results are deemed debits and negative calculation results are deemed credits.**  **When GLFSTMPLT3\_NEGATEEXPRESSION is TRUE and credits are deemed positive, positive calculation results are deemed credits and negative calculation results are deemed debits.** |
| **Total policy for column** | **GLFSTMPLT3\_TOTALPOLICY** | **Picklist** | **Add to totals** | **The calculation adds to totals like any non-calculated line. Totals are output as for non-calculated lines.** |
| **Calculate total lines** | **Do not accumulate totals. Continue to perform the defined calculation in total lines.** |
| **Null total lines** | **Return NULL in total lines.** |
| **Add to totals, then null line** | **Add the results of the calculation to totals but output NULL on non-total lines** |
| **Calculate total lines, null other lines** | **Perform the defined calculation on total lines but output NULL on non-total lines** |
|  | **EXTERNAL\_ID** |  |  | **GLFSTMPLT3\_GLFSTMPLT2 + GLFSTMPLT3\_FORMAT** |

*Table 3 –* ***GLFSTMPLT3***

Table **GLFSTMPLT4** – Financial statement template table # 4 – statement line header.

| **Field label** | **Column** | **Type/size** | **Possible values** | **Meaning / comment** |
| --- | --- | --- | --- | --- |
|  | **GLFSTMPLT4\_GLFSTMPLT2** | **Master/detail to GLFSTMPLT2** | **-** |  |
| **Line number** | **GLFSTMPLT4\_LINENO** | **Number(4,0)** | **-** | **Line # (major)** |
| **Non-printing line flag** | **GLFSTMPLT4\_NONPRINTING** | **Checkbox** | **-** | **Defaults to FALSE. Can be overridden if GLFSTMPLT4\_COMMAND = Accumulation line or Calculation line. If set to TRUE, this line is of a type that normally prints but this line is not printed. Its purpose is simply to accumulate totals.** |
| **Do not add line results to totals** | **GLFSTMPLT4\_NOTOTAL** | **Checkbox** | **-** | **Defaults to FALSE. Can be overridden if GLFSTMPLT4\_COMMAND = Accumulation line or Calculation line. If set to TRUE, the value of this line will be excluded from the running totals.** |
| **Line command** | **GLFSTMPLT4\_COMMAND** | **Picklist** | **-** | **Defines the purpose of the line:** |
| **Show debits as positive** | **Debits will print as positive numbers, credits will print as negative numbers, until a "Show credits as positive" command is encountered. For balance sheets and balance sheet schedules, this is the default value until a "Show credits as positive"command is read.** |
| **Show credits as positive** | **Credits will print as positive numbers, debits will print as negative numbers, until a "Show debits as positive" command is encountered. For P & Ls and P & L schedules, this is the default value unti a "Show debits as positive" command is read.** |
| **Change active total group** | **There are nine possible groups of totals, each group containing nine total levels. Most reports use only total group 1, which is the default setting until a "Change active total group" command is read.** |
| **Do a page break** | **Print a page break.** |
| **Print a blank line** | **Print a blank line.** |
| **Print a single underline** | **Print a single underline.** |
| **Print a double underline** | **Print a double underline.** |
| **Define a line name** | **Define and print (in column zero) a line name only.** |
| **Total level 1** | **Define line text and print and clear total level 1 in the active total group.** |
| **Total level 2** | **Define line text and print total level 2 and clear total levels 1 and 2 in the active total group.** |
| **Total level 3** | **Define line text and print total level 3 and clear total levels 1 through 3 in the active total group.** |
| **Total level 4** | **Define line text and print total level 4 and clear total levels 1 through 4 in the active total group.** |
| **Total level 5** | **Define line text and print total level 5 and clear total levels 1 through 5 in the active total group.** |
| **Total level 6** | **Define line text and print total level 6 and clear total levels 1 through 6 in the active total group.** |
| **Total level 7** | **Define line text and print total level 7 and clear total levels 1 through 7 in the active total group.** |
| **Total level 8** | **Define line text and print total level 8 and clear total levels 1 through 8 in the active total group.** |
| **Total level 9** | **Define line text and print total level 9 and clear total levels 1 through 9 in the active total group.** |
| **Accumulation line** | **Line includes text (unless GLFSTMPLT4\_NONPRINTING is TRUE) and accumulation instructions in GLFSTMPLT5.** |
| **Calculation line** | **Line includes text (unless GLFSTMPLT4\_NONPRINTING is TRUE) and calculation instructions below.** |
| **New total group** | **GLFSTMPLT4\_NEWTOTALGROUP** | **Number(1,0)** | **1 - 9** | **If GLFSTMPLT4\_COMMAND = Change active total group, this is the group number to which the active total group is being changed** |
| **Printed line text** | **GLFSTMPLT4\_TEXT** | **Text(50)** | **-** | **If GLFSTMPLT4\_COMMAND is not Define a line name, Accumulation line, Calculation line, or Total level *x*, this is NULL. Otherwise, it is the text to print in column zero when the line is output.** |
| **Indent text this many tabs** | **GLFSTMPLT4\_TEXTINDENT** | **Number(1,0)** | **0, 1, 2, or 3** | **Number of <TAB>s to indent GLFSTMPLT4\_TEXT from the left edge of column zero. Example of usage:  Fixed Assets  Less: reserve for depreciation** |
| **Bold the text** | **GLFSTMPLT4\_TEXTBOLD** | **Checkbox** |  | **NULL if GLFSTMPLT4\_TEXT is NULL, otherwise defaults to FALSE. If overridden to TRUE, GLFSTMPLT4\_TEXT will be bolded in the output.** |
| **Italicize the text** | **GLFSTMPLT4\_TEXTITALICS** | **Checkbox** |  | **NULL if GLFSTMPLT4\_TEXT is NULL, otherwise defaults to FALSE. If overridden to TRUE, GLFSTMPLT4\_TEXT will be italicized in the output.** |
| **Underline the text** | **GLFSTMPLT4\_TEXTUNDERLINE** | **Checkbox** |  | **NULL if GLFSTMPLT4\_TEXT is NULL, otherwise defaults to FALSE. If overridden to TRUE, GLFSTMPLT4\_TEXT will be underlined in the output.** |
| **Print line results as percentages** | **GLFSTMPLT4\_PRINTASPERCENTAGE** | **Checkbox** | **-** | **Always NULL unless GLFSTMPLT4\_COMMAND = Accumulation line, Calculation line, or Total level x. Otherwise, defaults to FALSE, meaning that column values will print in currency format unless overridden by a TRUE GLFSTMPLT3\_PRINTASPERCENTAGE. If set to TRUE, column values will print in percentage format. A TRUE GLFSTMPLT4\_PRINTASPERCENTAGE has precedence over a FALSE GLFSTMPLT3\_PRINTASPERCENTAGE. A TRUE GLFSTMPLT3\_PRINTASPERCENTAGE has precedence over a FALSE GLFSTMPLT4\_PRINTASPERCENTAGE.** |
| **Line label for reference by calculations** | **GLFSTMPLT4\_CALCLABEL** | **Text(50)** | **-** | **Always NULL unless GLFSTMPLT4\_COMMAND = Accumulation line, Calculation line, or Total level *x*. If defined, it should equal some GLFSTMPLT3\_VALUE1CALCLABEL or some GLFSTMPLT3\_VALUE2CALCLABEL. However, note that multiple columnar formats (GLFSTMPLT2\_FORMATs) can exist for the same set of report lines, so not every GLFSTMPLT4\_CALCLABEL will necessarily find a corresponding GLFSTMPLT3\_VALUE*x*CALCLABEL in every GLFSTMPLT2\_FORMAT. Therefore:  (1) GLFSTMPLT3\_VALUE*x*CALCLABLE without matching GLFSTMPLT4\_CALCLABEL is a fatal error.  (2) GLFSTMPLT4\_CALCLABEL without matching GLFSTMPLT3\_VALUExCALCLABLE is not.** |
|  |  |  |  | **The following fields define filters that apply to the whole report line.  NOTE: it is perfectly possibleto define a line whose potential contents are entirely filtered out by the report-level filters defined in GLFSTMPLT1. In the case of conflicting report-level (GLFSTMPLT1) and line-level (GLFSTMPLT4) definitions, the report level takes precedence.  It is also perfectly possible for the line filters defined in GLFSTMPLT4 to conflict with the column filters defined in GLFSTMPLT3. In that case, the intersection of the conflicting column and line will be null.** |
| **Filter on division** | **GLFSTMPLT4\_FILTERONDIV** | **Checkbox** |  | **Default value is FALSE. If set to TRUE, GLFSTMPLT4\_FILTERONPROJ can also be set to TRUE.** |
| **Filter on project** | **GLFSTMPLT4\_FILTERONPROJ** | **Checkbox** |  | **Default value is FALSE. If GLFSTMPLT4\_FILTERONDIV is set to TRUE, GLFSTMPLT4\_FILTERONPROJ can also be set to TRUE. If GLFSTMPLT4\_FILTERONDIV is set to FALSE, GLFSTMPLT4\_FILTERONPROJ must also be set to FALSE (since projects are assigned to divisions by definition).** |
| **Filter on dimension** | **GLFSTMPLT4\_FILTERONDIM** | **Checkbox** |  | **Default value is FALSE. Can be set to TRUE regardless of the values of GLFSTMPLT4\_FILTERONDIV and GLFSTMPLT4\_FILTERONPROJ.** |
| **Division filtering method** | **GLFSTMPLT4\_DIVMETHOD** | **Picklist** | **NULL** | **If GLFSTMPLT4\_FILTERONDIV is FALSE.** |
| **Variable division** | **User will enter division for this line at runtime.** |
| **Constant division** | **Division for this line will be defined in GLFSTMPLT4\_DIV.** |
| **No division** | **This value cannot be selected if GLFSTMPLT4\_FILTERONPROJ is TRUE because projects are divsion-specific.  Line values retrieved from GLAYSTMT for an account with have a division-specific GLAYSTMT\_GLSOURCE (corresponding to GLAYPERPROJ, GLAYPERDIV, GLAYPERDIMPROJ, or GLAYPERDIMDIV), where division is NULL. This will return only values where no division was entered on the originating transaction (in GLTXN). This is not at all the same as setting GLFSTMPLT4\_FILTERONDIV to FALSE - that will return account-wide values, regardless of whether a division was entered on the origination transaction or not.** |
| **Constant division for filtering** | **GLFSTMPLT4\_DIV** | **Lookup to GLDIV** |  | **NULL unless GLFSTMPLT3\_DIVMETHOD is Constant division. Otherwise, this is the division for this column.** |
| **Project filtering method** | **GLFSTMPLT4\_PROJMETHOD** | **Picklist** | **NULL** | **If GLFSTMPLT4\_FILTERONPROJ is FALSE.** |
| **Variable project** | **User will enter project for this line at runtime.** |
| **Constant project** | **Project for this line will be defined in GLFSTMPLT3\_PROJ.** |
| **No project** | **Line values retrieved from GLAYSTMT for an account with have a project-specific GLAYSTMT\_GLSOURCE (corresponding to GLAYPERPROJ or GLAYPERDIMPROJ), where project is NULL. This will return only values where no project was entered on the originating transaction (in GLTXN). This is not at all the same as setting GLFSTMPLT4\_FILTERONPROJ to FALSE - that will return account-wide values, regardless of whether a project was entered on the origination transaction or not.** |
| **Constant project for filtering** | **GLFSTMPLT4\_PROJ** | **Lookup to GLPROJ** |  | **NULL unless GLFSTMPLT4\_CONSTANTPROJ is TRUE. Otherwise, this is the project for this column.** |
| **Dimension filtering method** | **GLFSTMPLT4\_DIMMETHOD** | **Picklist** | **No dimension** | **Cancels the report-level dimension filter for one line only.** |
| **Constant dimension, constant value** | **Dimension # is always GLFSTMPLT4\_DIMNUM and dimension value is always GLFSTMPLT4\_DIMVAL.** |
| **Constant dimension, exclude constant value** | **Dimension # is always GLFSTMPLT4\_DIMNUM and dimension value is always GLFSTMPLT4\_DIMVAL.  This setting returns the value of all postings that reference GLFSTMPLT4\_DIMNUM except those that specify GLFSTMPLT4\_DIMVAL.** |
| **Constant dimension, variable value** | **Dimension # is always GLFSTMPLT4\_DIMNUM and dimension value is entered at runtime.** |
| **Constant dimension, exclude variable value** | **Dimension # is always GLFSTMPLT4\_DIMNUM and dimension value is entered at runtime.  This setting returns the value of all postings that reference GLFSTMPLT4\_DIMNUM except those that specify the dimension value entered at runtime.** |
| **Constant dimension, all values** | **Dimension # is always GLFSTMPLT4\_DIMNUM. This setting returns the sum of all postings where GLFSTMPLT4\_DIMNUM is specified.** |
| **Variable dimension, variable value** | **Dimension # and dimension value are entered at runtime.** |
| **Variable dimension, exclude variable value** | **Dimension # and dimension value are entered at runtime.  This setting returns the value of all postings that reference the dimension numnber entered at runtime except those that specify the dimension value entered at runtime.** |
| **Variable dimension, all values** | **Dimension # is entered at runtime. This setting returns the sum of all postings where the entered dimension # is specified.** |
| **All dimensions, all values** | **This setting returns the sum of all postings where any dimension # is specified.** |
| **No dimension** | **Values retrieved from GLAYSTMT for an account with have a dimension-specific GLAYSTMT\_GLSOURCE (corresponding to GLAYPERDIM, GLAYPERDIMDIV, or GLAYPERDIMPROJ, depending on the values of GLFSTMPLT4\_FILTERONDIM and GLFSTMPLT4\_FILTERONPROJ), where dimension is NULL. This will return only values where no dimension was entered on the originating transaction (in GLTXN).   This is not at all the same as setting GLFSTMPLT4\_FILTERONDIM to FALSE - that will return account-wide or account+division-wide values, regardless of whether a dimension was entered on the origination transaction or not.** |
| **Constant dimension for filtering** | **GLFSTMPLT4\_DIMNUM** | **Number(2,0)** | **Dimension #** | **NULL unless GLFSTMPLT4\_FILTERONDIM is TRUE and GLFSTMPLT4\_DIMMETHOD calls for a constant dimension #.** |
| **Constant dimension value for filtering** | **GLFSTMPLT4\_DIMVAL** | **Text(20)** | **Dimension value** | **NULL unless GLFSTMPLT4\_FILTERONDIM is TRUE and GLFSTMPLT4\_DIMMETHOD calls for a constant dimension value.** |
|  |  |  |  | **The following columns are NULL unless GLFSTMPLT4\_COMMAND = Clalculation line:** |
| **Expression operator** | **GLFSTMPLT4\_OPERATOR** | **Picklist** | **Plus Minus Multiplied by Divided by As a percentage of No operator** | **If GLFSTMPLT4\_OPERATOR = Plus,   then the value of the column = VALUE1 + VALUE2. If GLFSTMPLT4\_OPERATOR = Minus,   then the value of the column = VALUE1 - VALUE2. If GLFSTMPLT4\_OPERATOR = Multiplied by,   then the value of the column = VALUE1 \* VALUE2. If GLFSTMPLT4\_OPERATOR = Divided by,   then the value of the column = VALUE1 / VALUE2. If GLFSTMPLT4\_OPERATOR = As a percentage of,   then the value of the column = (VALUE1 / VALUE2) \* 100. If GLFSTMPLT4\_OPERATOR = No operator,  then the value of the column = VALUE1.** |
| **Origin of equation left side** | **GLFSTMPLT4\_VALUE1TYPE** | **Picklist** | **Other row, same column Other row, other column Constant Entered at runtime Imported at runtime Special** | **If Other row, same column  VALUE1 will be set the value of the row in the same column whose GLFSTMPLT4\_CALCLABEL  = this row's GLFSTMPLT4\_VALUE1CALCLABEL. If Other row, other column  VALUE1 will be set the value of the row in GLFSTMPLT4\_VALUE1COL whose   GLFSTMPLT4\_CALCLABEL = this row's GLFSTMPLT4\_VALUE1CALCLABEL. If Constant,  VALUE1 for each row will be set to GLFSTMPLT4\_VALUE1CONSTANT.  If Entered at runtime,  VALUE1 for all rows will be entered at runtime.  If Imported at runtime,  VALUE1 for all rows will be read from GLFSTMPLT6 where   GLFSTMPLT6\_CMPNO = GLFSTMPL1\_CMPNO,   GLFSTMPLT6\_YEAR = current year,   GLFSTMPLT6\_PERIOD = current period,   GLFSTMPLT6\_GLDIV = division of column scope (if any),  GLFSTMPLT6\_GLPROJ = project of column scope (if any),  GLFSTMPLT6\_DIMNUM = dimension # of column scope (if any),  GLFSTMPLT6\_DIMVAL = dimension value of column scope (if any), and  GLFSTMPLT6\_LABEL = GLFSTMPLT3\_VALUE1CALCLABEL. If Special,  VALUE1 is determined by GLFSTMPLT3\_VALUE1SPECIALTYPE.  If Other row, same column or Total, same column, VALUE1 will vary from column to column. Otherwise, VALUE1 is the same across all columns.** |
| **Origin of equation right side** | **GLFSTMPLT4\_VALUE2TYPE** | **Picklist** | **Other row, same column Other row, other column Constant Entered at runtime Imported at runtime Special** | **If Other row, same column  VALUE2 will be set the value of the row in the same column whose GLFSTMPLT4\_CALCLABEL  = this row's GLFSTMPLT4\_VALUE2CALCLABEL. If Other row, other column  VALUE2 will be set the value of the row in GLFSTMPLT4\_VALUE2COL whose   GLFSTMPLT4\_CALCLABEL = this row's GLFSTMPLT4\_VALUE2CALCLABEL.  If Constant,  VALUE2 for each row will be set to GLFSTMPLT4\_VALUE2CONSTANT.  If Entered at runtime,  VALUE2 for all rows will be entered at runtime.  If Imported at runtime,  VALUE2 for all rows will be read from GLFSTMPLT6 where   GLFSTMPLT6\_CMPNO = GLFSTMPL1\_CMPNO,   GLFSTMPLT6\_YEAR = current year,   GLFSTMPLT6\_PERIOD = current period,   GLFSTMPLT6\_GLDIV = division of column scope (if any),  GLFSTMPLT6\_GLPROJ = project of column scope (if any),  GLFSTMPLT6\_DIMNUM = dimension # of column scope (if any),  GLFSTMPLT6\_DIMVAL = dimension value of column scope (if any), and  GLFSTMPLT6\_LABEL = GLFSTMPLT3\_VALUE2CALCLABEL. If Special,  VALUE12is determined by GLFSTMPLT3\_VALUE2SPECIALTYPE.  If Other row, same column or Total, same column, VALUE2 will vary from column to column. Otherwise, VALUE2 is the same across all columns.** |
| **Constant value of equation left side** | **GLFSTMPLT4\_VALUE1CONSTANT** | **Number(16,2)** | **-** | **NULL unless GLFSTMPLT4\_VALUE1TYPE = Constant. Otherwise, this is the value of VALUE1.** |
| **Constant value of equation right side** | **GLFSTMPLT4\_VALUE2CONSTANT** | **Number(16,2)** | **-** | **NULL unless GLFSTMPLT4\_VALUE2TYPE = Constant. Otherwise, this is the value of VALUE2.** |
| **Type of special value for equation left side** | **GLFSTMPLT4\_VALUE1SPECIALTYPE** | **Picklist** | **Period number Periods in year** | **NULL unless GLFSTMPLT4\_VALUE1TYPE = Special. If Period number,  VALUE1 = the current period number. If Periods in year,  VALUE1 = the number of periods in the year.  Examples of use:  To average the period values in the current year,   divide a year to date column by the period number.  To project the year-end balance from a single period,  multiply a column for that period by (periods in the current year - period number + 1)   and add it to the current year to date.** |
| **Type of special value for equation right side** | **GLFSTMPLT4\_VALUE2SPECIALTYPE** | **Picklist** | **Period number Periods in year** | **NULL unless GLFSTMPLT4\_VALUE2TYPE = Special. If Period number,  VALUE2 = the current period number. If Periods in year,  VALUE2 = the number of periods in the year.** |
| **Line label for source of equation left side** | **GLFSTMPLT4\_VALUE1CALCLABEL** | **Text(50)** | **-** | **NULL unless GLFSTMPLT4\_VALUE1TYPE = Other row, other column or Imported at runtime. If GLFSTMPLT4\_VALUE1TYPE = Other row, other column, this label must equal some line label defined as GLFSTMPLT4\_CALCLABEL in GLFSTMPLT4. If GLFSTMPLT4\_VALUE1TYPE = Imported at runtime, this label is used to look up VALUE1 in GLFSTMPLT6.  Typical values of calculation labels, suggesting their usage:  For line labels:  NET\_INCOME  TOTAL\_COST\_OF\_SALES  TOTAL\_G\_S\_AND\_A  For imported data labels:  SQ\_FEET\_IN\_PLANT, SQ\_FEET\_IN\_DEPT  HEADCOUNT\_FOR\_COMPANY, HEADCOUNT\_FOR\_DIVISION  KILOGRAMMES\_SHIPPED\_IN\_PERIOD  If there are unresolved line labels - e.g. GLFSTMPLT4\_VALUE1CALCLABEL is not NULL but no corresponding GLFSTMPLT4\_CALCLABEL exists - the template is unusable, the user is warned, and GLFSTMPLT2\_UNRESOLVEDLABELS is set to TRUE - meaning that statement calculation cannot be run for GLFSTMPLT2\_FORMAT** |
| **Line label for source of equation right side** | **GLFSTMPLT4\_VALUE2CALCLABEL** | **Text(50)** | **-** | **NULL unless GLFSTMPLT4\_VALUE2TYPE = Other row, other column or Imported at runtime. If GLFSTMPLT4\_VALUE2TYPE = Other row, other column, this label must equal some line label defined as GLFSTMPLT4\_CALCLABEL in GLFSTMPLT4. If GLFSTMPLT4\_VALUE2TYPE = Imported at runtime, this label is used to look up VALUE2 in GLFSTMPLT6.** |
| **Column for source of equation left side** | **GLFSTMPLT4\_VALUE1COL** | **Number(2,0)** | **-** | **NULL unless GLFSTMPLT4\_VALUE1TYPE = Other row, other column. Otherwise this is the column in which VALUE1 will be found at runtime.** |
| **Column for source of equation right side** | **GLFSTMPLT4\_VALUE2COL** | **Number(2,0)** | **-** | **NULL unless GLFSTMPLT4\_VALUE2TYPE = Other row, other column. Otherwise this is the column in which VALUE2 will be found at runtime.** |
| **Negate results of expression** | **GLFSTMPLT4\_NEGATEEXPRESSION** | **Checkbox** | **-** | **Defaults to FALSE. If set to TRUE, the sign of the calculation result is flipped (the result is multiplied by -1).  This inverts the default behavior for displaying and totaling calculation results:   When GLFSTMPLT4\_NEGATEEXPRESSION is FALSE and debits are deemed positive (see GLFSTMPLT4\_COMMAND), positive calculation results are deemed debits and negative calculation results are deemed credits.  When GLFSTMPLT4\_NEGATEEXPRESSION is FALSE and credits are deemed positive, positive calculation results are deemed credits and negative calculation results are deemed debits.  When GLFSTMPLT4\_NEGATEEXPRESSION is TRUE and debits are deemed positive, positive calculation results are deemed debits and negative calculation results are deemed credits.  When GLFSTMPLT4\_NEGATEEXPRESSION is TRUE and credits are deemed positive, positive calculation results are deemed credits and negative calculation results are deemed debits.** |
| **First label for external data exported from this line** | **GLFSTMPLT4\_EXPORTLABEL1** | **Text(50)** | **-** | **NULL by default. GLFSTMPLT4\_EXPORTLABEL1 through GLFSTMPLT4\_EXPORTLABEL8 can be assigned a value if GLFSTMPLT4\_COMMAND is Accumulation line, Calculation line, or any total line. It is pointless for more than one line to share a GLFSTMPLT4\_EXPORTLABEL*n*, since the results of the higher-numbered line will simply overwrite the results of the lower-numbered line. If, for any intersection of line and column, GLFSTMPLT4\_COLEXPORTLABEL*n* = GLFSTMPLT3\_EXPORTLABEL, the value of the intersection is written to GLFSTMPLT6 at report generation with:  GLFSTMPLT6\_CMPNO = GLFSTMPLT1\_CMPNO  GLFSTMPLT6\_LABEL = GLFSTMPLT4\_EXPORTLABEL*n* (where *n* corresponds to   the *n* in GLFSTMPLT4\_COLEXPORTLABEL*n*)  GLFSTMPLT6\_FROMYEAR = Column year  GLFSTMPLT6\_FROMPERIOD = Column period  GLFSTMPLT6\_THRUYEAR = Column year  GLFSTMPLT6\_THRUPERIOD = Column period  GLFSTMPLT6\_SCALE = Currency  GLFSTMPLT6\_CURRENCYVAL = line/column value  GLFSTMPLT6\_GENERALVAL = NULL** |
| **Second label for external data exported from this line** | **GLFSTMPLT4\_EXPORTLABEL2** | **Text(50)** | **-** | **See GLFSTMPLT4\_EXPORTLABEL1** |
| **Third label for external data exported from this line** | **GLFSTMPLT4\_EXPORTLABEL3** | **Text(50)** | **-** | **See GLFSTMPLT4\_EXPORTLABEL1** |
| **Fourth label for external data exported from this line** | **GLFSTMPLT4\_EXPORTLABEL4** | **Text(50)** | **-** | **See GLFSTMPLT4\_EXPORTLABEL1** |
| **Fifth label for external data exported from this line** | **GLFSTMPLT4\_EXPORTLABEL5** | **Text(50)** | **-** | **See GLFSTMPLT4\_EXPORTLABEL1** |
| **Sixth label for external data exported from this line** | **GLFSTMPLT4\_EXPORTLABEL6** | **Text(50)** | **-** | **See GLFSTMPLT4\_EXPORTLABEL1** |
| **Seventh label for external data exported from this line** | **GLFSTMPLT4\_EXPORTLABEL7** | **Text(50)** | **-** | **See GLFSTMPLT4\_EXPORTLABEL1** |
| **Eighth label for external data exported from this line** | **GLFSTMPLT4\_EXPORTLABEL8** | **Text(50)** | **-** | **See GLFSTMPLT4\_EXPORTLABEL1** |
| **First label to be matched to a GLFSTMPLT3\_EXPORTLABEL** | **GLFSTMPLT4\_COLEXPORTLABEL1** | **Text(50)** | **-** | **NULL by default. GLFSTMPLT4\_EXPORTLABEL1 through GLFSTMPLT4\_EXPORTLABEL8 can be assigned a value if GLFSTMPLT4\_COMMAND is Accumulation line, Calculation line, or any total line. It is pointless for more than one line to share a GLFSTMPLT4\_EXPORTLABEL*n*, since the results of the higher-numbered line will simply overwrite the results of the lower-numbered line. If, for any intersection of line and column, GLFSTMPLT4\_COLEXPORTLABEL*n* = GLFSTMPLT3\_EXPORTLABEL, the value of the intersection is written to GLFSTMPLT6 at report generation with:  GLFSTMPLT6\_CMPNO = GLFSTMPLT1\_CMPNO  GLFSTMPLT6\_LABEL = GLFSTMPLT4\_EXPORTLABEL*n* (where *n* corresponds to   the *n* in GLFSTMPLT4\_COLEXPORTLABEL*n*)  GLFSTMPLT6\_FROMYEAR = Column year  GLFSTMPLT6\_FROMPERIOD = Column period  GLFSTMPLT6\_THRUYEAR = Column year  GLFSTMPLT6\_THRUPERIOD = Column period  GLFSTMPLT6\_SCALE = Currency  GLFSTMPLT6\_CURRENCYVAL = line/column value  GLFSTMPLT6\_GENERALVAL = NULL** |
| **Second label to be matched to a GLFSTMPLT3\_EXPORTLABEL** | **GLFSTMPLT4\_COLEXPORTLABEL2** | **Text(50)** | **-** | **See GLFSTMPLT4\_COLEXPORTLABEL1** |
| **Third label to be matched to a GLFSTMPLT3\_EXPORTLABEL** | **GLFSTMPLT4\_COLEXPORTLABEL3** | **Text(50)** | **-** | **See GLFSTMPLT4\_COLEXPORTLABEL1** |
| **Fourth label to be matched to a GLFSTMPLT3\_EXPORTLABEL** | **GLFSTMPLT4\_COLEXPORTLABEL4** | **Text(50)** | **-** | **See GLFSTMPLT4\_COLEXPORTLABEL1** |
| **Fifth label to be matched to a GLFSTMPLT3\_EXPORTLABEL** | **GLFSTMPLT4\_COLEXPORTLABEL5** | **Text(50)** | **-** | **See GLFSTMPLT4\_COLEXPORTLABEL1** |
| **Sixth label to be matched to a GLFSTMPLT3\_EXPORTLABEL** | **GLFSTMPLT4\_COLEXPORTLABEL6** | **Text(50)** | **-** | **See GLFSTMPLT4\_COLEXPORTLABEL1** |
| **Seventh label to be matched to a GLFSTMPLT3\_EXPORTLABEL** | **GLFSTMPLT4\_COLEXPORTLABEL7** | **Text(50)** | **-** | **See GLFSTMPLT4\_COLEXPORTLABEL1** |
| **Eighth label to be matched to a GLFSTMPLT3\_EXPORTLABEL** | **GLFSTMPLT4\_COLEXPORTLABEL8** | **Text(50)** | **-** | **See GLFSTMPLT4\_COLEXPORTLABEL1** |

*Table 4 –* ***GLFSTMPLT4***

Table **GLFSTMPLT5** – Financial statement template table # 5 – statement line calculation detail.

| **Field label** | **Column** | **Type/size** | **Possible values** | **Meaning / comment** |
| --- | --- | --- | --- | --- |
|  | **GLFSTMPLT5\_GLFSTMPLT4** | **Master/detail to GLFSTMPLT4** | **-** |  |
| **Subline sequence #** | **GLFSTMPLT5\_LINESEQ** | **Number(4,0)** | **> 0** | **Sequence within line. Commands re executed in GLFSTMPLT5\_LINESEQ order. If only add and subtract commands are used, the order of execution doesn't matter. When multiple and divide commands are used, the line value depends on the order of execution.** |
| **Subline command** | **GLFSTMPLT5\_COMMAND** | **Picklist** | **Comment** | **The subline is a comment, which is entered into GLFSTMPLT5\_TEXT. The most common use of subline columns occurs when sublines whose GLFSTMPLT5\_COMMAND = Add an account or Subtract an account are cloned from one company to another. If GLFSTMPLT5\_GLACCT does not exist for the new company, the subline is changed to a comment that identifies the old GLFSTMPLT5\_COMMAND and GLFSTMPLT5\_GLACCT. Likewise, if the old GLFSTMPLT5\_COMMAND = Invoke a predefined subline group, the new subline is changed to a comment showing the GLFSTMPLT5\_COMMAND and the predefined subline group identified in GLFSTMPLT5\_GLFSTMPLT7.** |
| **Add an account** | **Add one account, to be named in GLFSTMPLT5\_GLACCT. Account must exist in GLACCT.** |
| **Subtract an account** | **Subtract one account, to be named in GLFSTMPLT5\_GLACCT. Account must exist in GLACCT.** |
| **Add an account range** | **Add a range of accounts, as specified by GLFSTMPLT5\_RANGESTART and GLFSTMPLT5\_RANGEEND. Accounts need not actually be defined in GLACCT.** |
| **Subtract an account range** | **Subtract a range of accounts, as specified by GLFSTMPLT5\_RANGESTART and GLFSTMPLT5\_RANGEEND. Accounts need not actually be defined in GLACCT.** |
| **Invoke a predefined subline group** | **The sublines in the predefined subline group identified by GLFSTMPLT5\_GLFSTMPLT7 will be executed in GLFSTMPLT8\_SEQ order starting at GLFSTMPLT5\_LINESEQ. After the last subline in the group has been executed, execution of sublines in GLFSTMPLT5 will resume with the next GLFSTMPLT5\_LINESEQ. The effect is exactly as though the sublines in the subline group had been defined in GLFSTMPLT5.** |
| **Add labeled line** | **Add the contents of a labeled line. GLFSTMPLT5\_SOURCELINE must be defined as line label in GLFSTMPLT4.** |
| **Subtract labeled line** | **Subtract the contents of a labeled line. GLFSTMPLT5\_SOURCELINE must be defined as line label in GLFSTMPLT4.** |
| **Multiply by labeled line** | **Multiply by the contents of a labeled line. GLFSTMPLT5\_SOURCELINE must be defined as line label in GLFSTMPLT4.** |
| **Divide by labeled line** | **Divide by the contents of a labeled line. GLFSTMPLT5\_SOURCELINE must be defined as line label in GLFSTMPLT4.** |
| **Add imported data** | **Add data retrieved from GLFSTMPLT6. GLFSTMPLT5\_SOURCELINE must exist in GLFSTMPLT6.** |
| **Subtract imported data** | **Subtract data retrieved from GLFSTMPLT6. GLFSTMPLT5\_SOURCELINE must exist in GLFSTMPLT6.** |
| **Multiply by imported data** | **Multiply by data retrieved from GLFSTMPLT6. GLFSTMPLT5\_SOURCELINE must exist in GLFSTMPLT6.** |
| **Divide by imported data** | **Divide by data retrieved from GLFSTMPLT6. GLFSTMPLT5\_SOURCELINE must exist in GLFSTMPLT6.** |
|  |  |  |  | **Note 1: if a single account is specified (GLFSTMPLT5\_COMMAND = Add an account or Subtract account), then that account must exist. But if only a range of accounts is specified, the range start and end points needn't be real accounts. E.g. if the purpose of a line is to print total A/R and A/R accounts are defined as 12*nn*, then it is prudent to set the range start to "12" and the range end to "12ZZ."** |
|  |  |  |  | **Note 2: GLFSTMPLT5\_COMMANDs are executed in order of GLFSTMPLT5\_LINESEQ.  The sequence of GLFSTMPLT5\_COMMANDs is irrelevant if a GLFSTMPLT4 line contains only add and subtract commands.  However, the sequence matters if multiply and divide commands are used. Thus, the command sequence  Add 6  Add 8  Divide by 2  Add 10 Evaluates to ((6+8)/2)+10 = 17.  But the sequence  Add 6  Add 8  Add 10  Divide by 2 Evaluates to (6+8+10)/2 = 12.** |
| **Start of account range** | **GLFSTMPLT5\_RANGESTART** | **Text(30)** | **-** | **NULL unless GLFSTMPLT5\_COMMAND = Add an account range or Subtract an account range. Otherwise, lexically < or = to the first account in the range.** |
| **End of account range** | **GLFSTMPLT5\_RANGEEND** | **Text(30)** | **>= GLFSTMPLT5\_RANGESTART** | **NULL unless GLFSTMPLT5\_COMMAND = Add an account range or Subtract an account range. Otherwise, lexically > or = to the last account in the range.** |
| **Company for accounts** | **GLFSTMPLT5\_CMPNO** | **Lookup to GLCMP** | **Must be the reporting company or a subsidiary of the reporting company** | **Data from GLAYSTMT will be retrieved where GLFSTMPLT5\_CMPNO = GLAYSTMT\_SUBSIDIARY** |
| **Account number** | **GLFSTMPLT5\_GLACCT** | **Lookup to GLACCT** | **-** | **Required if GLFSTMPLT5\_COMMAND = Add an account or Subtract an account. NULL otherwise.** |
| **Label of line that is data source** | **GLFSTMPLT5\_SOURCELINE** | **Text(50)** | **-** | **For Add labeled line, Subtract labeled line, Multiply by labeled line, and Divide by labeled line commands, this is the GLFSTMPLT4\_CALCLABEL of the line containing the operand. For Add imported data, Subtract imported data, Multiply by imported data, and Divide by imported data, this is the GLFSTMPLT6\_LABEL that identifies the operand.  Note that when line labels are used, an operand for each column is retrieved and applied, but when imported data labels are used, a single operand is retrieved and applied to all columns.** |
| **Invoke this group of predefined sublines** | **GLFSTMPLT5\_GLFSTMPLT7** | **Lookup to GLFSTMPLT7** | **-** | **NULL unless GLFSTMPLT5\_COMMAND = Invoke a predefined subline group. Otherwise, this is the predefined subline group being invoked.** |
| **Comment** | **GLFSTMPLT5\_TEXT** | **Text(100)** | **-** | **Contains comment text when GLFSTMPLT5\_COMMAND = Comment.** |
|  | **EXTERNAL\_ID** |  |  | **GLFSTMPLT5\_GLFSTMPLT4 + GLFSTMPLT5\_LINESEQ** |

*Table 5 –* ***GLFSTMPLT5***

Table **GLFSTMPLT6** – Financial statement template table # 6 – external data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field label** | **Column** | **Type/size** | **Possible values** | **Meaning / comment** |
| **Company** | **GLFSTMPLT6\_CMPNO** | **Lookup to GLCMP** |  | **Set to GLFSTMPLT1\_CMPNO if GLFSTMPLT6\_LABEL if exported from statement.** |
| **External data label** | **GLFSTMPLT6\_LABEL** | **Text(50)** | **Not NULL** | **Matched to GLFSTMPLT*n*\_VALUE*v*CALCLABEL.** |
| **Start of effective year range** | **GLFSTMPLT6\_FROMYEAR** | **Lookup to GLYEAR** |  | **Year in which this record became valid** |
| **Start of effective period range** | **GLFSTMPLT6\_FROMPERIOD** | **Number(2,0)** |  | **Period within GLFSTMPLT6\_FROMYEAR in which this record became valid** |
| **End of effective year range** | **GLFSTMPLT6\_THRUYEAR** | **Lookup to GLYEAR** |  | **NULL if this record is still valid for current periods (it is always valid for prior period values). Else, the last year in which this record was valid for current periods.** |
| **End of effective period range** | **GLFSTMPLT6\_THRUPERIOD** | **Number(2,0)** |  | **NULL if GLFSTMPLT6\_THRUYEAR is NULL. Else, the last period GLFSTMPLT6\_THRUYEAR in which this record was valid.** |
| **Scale of decimal precision** | **GLFSTMPLT6\_SCALE** | **Picklist** | **Currency General** | **Default is Currency. Specify General if > 2 digits of precision are required. If Currency, use GLFSTMPLT6\_CURRENCYVAL else use GLFSTMPLT6\_GENERALVAL.** |
| **Currency amount** | **GLFSTMPLT6\_CURRENCYVAL** | **Number(16,2)** |  |  |
| **Non-currency amount** | **GLFSTMPLT6\_GENERALVAL** | **Number(10,8)** |  |  |
|  | **EXTERNAL ID** |  |  | **GLFSTMPLT6\_CMPNO + GLFSTMPLT6\_LABEL + GLFSTMPLT6\_FROMYEAR + GLFSTMPLT6\_FROMPERIOD.  In other words, it is legal for no record for a given label to be valid for a particular year + period, but it is not legal for more than one record for a given label to be in effect at any one time. If a record for a particular label is to be superseded, it must be closed (assigned the GLFSTMPLT6\_THRUEAR and GLFSTMPLT6\_THRUPERIOD in which it was last valid) before a new record for the label can be added, and the new record's GLFSTMPLT6\_FROMYEAR + GLFSTMPLT6\_FROMPERIOD must be at least one period later than the old record's GLFSTMPLT6\_THRUEAR + GLFSTMPLT6\_THRUPERIOD.** |

*Table 6 –* ***GLFSTMPLT6***

Table **GLFSTMPLT7** – Financial statement template table # 7 – predefined subline group header.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field label** | **Column** | **Type/size** | **Possible values** | **Meaning / comment** |
| **Company** | **GLFSTMPLT7\_GLCMP** | **Lookup to GLCMP** | **-** | **Reporting company** |
| **Predefined subline group** | **GLFSTMPLT7\_NAME** | **Text(50)** | **Not NULL** | **The name of the predefined subline group** |
|  | **EXTERNAL ID** |  |  | **GLFSTMPLT7\_GLCMP + GLFSTMPLT7\_NAME** |

*Table 7 –* ***GLFSTMPLT7***

Table **GLFSTMPLT8** – Financial statement template table # 8 – predefined subline group detail.

| **Field label** | **Column** | **Type/size** | **Possible values** | **Meaning / comment** |
| --- | --- | --- | --- | --- |
|  | **GLFSTMPLT8\_GLFSTMPLT7** | **Master/detail to GLFSTMPLT7** | **-** |  |
| **Subline sequence #** | **GLFSTMPLT8\_SEQ** | **Number(4,0)** | **-** | **Sequence within calculation. Commands re executed in GLFSTMPLT8\_SEQ order. If only add and subtract commands are used, the order of execution doesn't matter. When multiple and divide commands are used, the line value depends on the order of execution.** |
| **Subline command** | **GLFSTMPLT8\_COMMAND** | **Picklist** | **Add an account** | **Add one account, to be named in GLFSTMPLT8\_GLACCT. Account must exist in GLACCT.** |
| **Subtract an account** | **Subtract one account, to be named in GLFSTMPLT8\_GLACCT. Account must exist in GLACCT.** |
| **Add an account range** | **Add a range of accounts, as specified by GLFSTMPLT8\_RANGESTART and GLFSTMPLT8\_RANGEEND. Accounts need not actually be defined in GLACCT.** |
| **Subtract an account range** | **Subtract a range of accounts, as specified by GLFSTMPLT8\_RANGESTART and GLFSTMPLT8\_RANGEEND. Accounts need not actually be defined in GLACCT.** |
|  |  |  |  | **Note 1: if a single account is specified (GLFSTMPLT5\_COMMAND = Add an account or Subtract account), then that account must exist. But if only a range of accounts is specified, the range start and end points needn't be real accounts. E.g. if the purpose of a line is to print total A/R and A/R accounts are defined as 12*nn*, then it is prudent to set the range start to "12" and the range end to "12ZZ."** |
|  |  |  |  | **Note 2: GLFSTMPLT5\_COMMANDs are executed in order of GLFSTMPLT5\_LINESEQ.  The sequence of GLFSTMPLT5\_COMMANDs is irrelevant if a GLFSTMPLT4 line contains only add and subtract commands.  However, the sequence matters if multiply and divide commands are used. Thus, the command sequence  Add 6  Add 8  Divide by 2  Add 10 Evaluates to ((6+8)/2)+10 = 17.  But the sequence  Add 6  Add 8  Add 10  Divide by 2 Evaluates to (6+8+10)/2 = 12.** |
| **Start of account range** | **GLFSTMPLT8\_RANGESTART** | **Text(30)** | **-** | **NULL if GLFSTMPLT8\_COMMAND = Add an account or Subtract an account. Otherwise, lexically < or = to the first account in the range.** |
| **End of account range** | **GLFSTMPLT8\_RANGEEND** | **Text(30)** | **>= GLFSTMPLT8\_RANGESTART** | **NULL if GLFSTMPLT8\_COMMAND = Add an account or Subtract an account. Otherwise, lexically > or = to the last account in the range.** |
| **Company for accounts** | **GLFSTMPLT8\_CMPNO** | **Lookup to GLCMP** | **Must be the reporting company (GLFSTMPLT7\_GLCMP) or a subsidiary of the reporting company** | **This is not necessarily the same as GLFSTMPLT7\_GLCMP: it may be a subsidiary of GLFSTMPLT7\_GLCMP. Data from GLAYSTMT will be retrieved where GLFSTMPLT8\_CMPNO = GLAYSTMT\_SUBSIDIARY** |
| **Account number** | **GLFSTMPLT8\_GLACCT** | **Lookup to GLACCT** | **-** | **Required if GLFSTMPLT8\_COMMAND = Add an account or Subtract an account. NULL otherwise.** |
|  | **EXTERNAL\_ID** |  |  | **GLFSTMPLT8\_GLFSTMPLT7 + GLFSTMPLT8\_SEQ** |

*Table 8 –* ***GLFSTMPLT8***

Table **GLFSTMPLT9** – Export label list for enforcing uniqueness (export data labels in **GLFSTMPLT4** and **GLFSTMPLT6** must never be duplicated).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field label** | **Column** | **Type/size** | **Possible values** | **Meaning / comment** |
| **Company** | **GLFSTMPLT9\_GLCMP** | **Lookup to GLCMP** | **-** | **Reporting company** |
| **External data label** | **GLFSTMPLT9\_EXPORTLABEL** | **Text(50)** | **-** | **External label used as a GLFSTMPLT6\_LABEL.** |
| **Used in line** | **GLFSTMPLT9\_GLFSTMPLT4** | **Lookup to GLFSTMPLT4** | **-** | **Lookup to the one and only line in which GLFSTMPLT9\_EXPORTLABEL appears as a GLFSTMPLT4\_EXPORTLABEL*n*** |
|  | **EXTERNAL\_ID** |  |  | **GLFSTMPLT9\_GLCMP + GLFSTMPLT9\_EXPORTLABEL** |

*Table 9 –* ***GLFSTMPLT9***

Table **TMPLTMVR** – Export/import parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field label** | **Column** | **Type/size** | **Possible values** | **Meaning / comment** |
| **Copy 1 template or all for 1 company?** | **TMPLTMVR\_ONETMPLTORALL** | **Picklist** | **One, All** | ***One* means copy a single template. *All* means copy all templates for one company.** |
| **Copy sublines?** | **TMPLTMVR\_SUBLINES** | **Checkbox** | **-** | **If true, we expect to find all GL accounts referenced by GLFSTMPLT5 in the destination org. If we don't find one we'll terminate with an error.** |
| **Company record ID in source org** | **TMPLTMVR\_GLCMP** | **Lookup to GLCMP** | **-** |  |
| **Source GLCMP record name** | **TMPLTMVR\_GLCMPNSRCNAME** | **Text** | **-** | **Record name field of GLCMP in source org** |
| **Source template header record name** | **TMPLTMVR\_GLFSTMPLT1SRCNAME** | **Text** | **-** | **If TMPLTMVR\_ONETMPLTORALL = One, this is the record name field of the selected GLFSTMPLT1 record in the source org. If TMPLTMVR\_ONETMPLTORALL = All, this is null.** |

*Table 10 –* ***TMPLTMVR***

Table **TMPLTMVRID** – Record ID cross reference from source org to destination org.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field label** | **Column** | **Type/size** | **Possible values** | **Meaning / comment** |
| **Table name** | **TMPLTMVRID\_TABLE** | **Text** | **-** | **Table name** |
| **Table field name** | **TMPLTMVRID\_FIELDNAME** | **Text** | **-** | **Field name in TMPLTMVRID\_TABLE** |
| **Record's name field** | **TMPLTMVRID\_NAME** | **Text** |  | **Name field for TMPLTMVRID\_FIELDNAME in TMPLTMVRID\_TABLE record** |
| **Record ID in source org** | **TMPLTMVRID\_SRCID** | **Text** |  | **Record ID in source org** |
| **Record ID in dest org** | **TMPLTMVRID\_DESTID** | **Text** |  | **Record ID in destination org** |
|  | **External ID** |  |  | **TMPLTMVRID\_TABLE + TMPLTMVRID\_FIELDNAME + TMPLTMVRID\_SRCID** |

*Table 11 –* ***TMPLTMVRID***

Table **TMPLTCOA** – Source org chart of accounts to destination org chart of accounts translation table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field label** | **Column** | **Type/size** | **Possible values** | **Meaning / comment** |
| **Account NAME field** | **TMPLTCOA\_GLACCTNAME** | **Text** | **-** | **NAME field in GLACCT** |
| **Source record ID** | **TMPLTCOA\_SRCID** | **Text** | **-** | **Record ID in source GLACCT** |
| **Destination record ID** | **TMPLTCOA\_DESTID** | **Text** | **-** | **Record ID in destination GLACCT** |
|  | **External ID** |  |  | **TMPLTCOA\_SRCID** |

*Table 12 –* ***TMPLTCOA***

Table **TEMPTMPLT1** – Temporary version of **GLFSTMPLT1** – top-level header.

| **Field label** | **Column** | **Type/size** | **Possible values** | **Meaning / comment** |
| --- | --- | --- | --- | --- |
| **Reporting company** | **TEMPTMPLT1\_CMPNO** | **Lookup to GLCMP** | **-** | **Translate to new record ID when writing new GLFSTMPLT1.** |
| **Template ID** | **TEMPMPLT1\_TEMPLATE** | **Text(30)** | **-** | **Identifies the report being specified by the template.** |
| **Report type** | **TEMPTMPLT1\_REPORTTYPE** | **Picklist** | **Balance sheet** | **Formal balance sheet** |
| **Balance sheet schedule** | **Supporting schedule to balance sheet** |
| **P & L** | **Formal P & L** |
| **P & L schedule** | **Supporting schedule to P & L** |
| **Cash flow statement** | **Cash flow statement (aka statement of changes to financial position). Not supported in initial release.** |
| **Cash flow stmt schedule** | **Supporting schedule to cash flow statement. Not supported in initial release.** |
| **Report scope** | **TEMPTMPLT1\_SCOPE** | **Picklist** | **Company wide Single division Single project Company plus dimension Division plus dimension Project plus dimension** | **Must be Company wide if GLFSTMPLT1\_REPORTTYPE = Balance sheet. Must be Company wide or Company plus dimension if GLFSTMPLT1\_REPORTTYPE = Balance sheet schedule.** |
| **Define a constant division for filtering** | **TEMPTMPLT1\_CONSTANTDIV** | **Checkbox** | **-** | **NULL unless GLFSTMPLT1\_SCOPE = Single division, Single project, Division plus dimension, or Project plus dimension. If TRUE, this template only applies to GLFSTMPLT1\_DIV. If FALSE, user will be prompted for division at run time.** |
| **Define a constant project for filtering** | **TEMPTMPLT1\_CONSTANTPROJ** | **Checkbox** | **-** | **NULL unless GLFSTMPLT1\_SCOPE = Single project or Project plus dimension. NULL if GLFSTMPLT1\_REPORTTYPE = Balance sheet. If TRUE, this template only applies to GLFSTMPLT1\_PROJ. If FALSE, user will be prompted for project at run time.** |
| **Filter whole report on a single dimension** | **TEMPTMPLT1\_FILTERONDIMNUM** | **Checkbox** | **-** | **NULL unless GLFSTMPLT1\_SCOPE = Company plus dimensions or Single div plus dimensions or Single proj plus dimensions. If TRUE, the whole report will be confined to a single dim #. If FALSE, use of dimensions (if any) is confined to column definitions (in GLFSTMPLT3).** |
| **Filter whole report on a value for a single dimension** | **TEMPTMPLT1\_FILTERONDIMVAL** | **Checkbox** | **-** | **NULL unless GLFSTMPLT1\_FILTERONDIMNUM is TRUE. If TRUE, the whole report will be confined to a single pair of dim # and dim value. If FALSE, the report will be confined to a single dim # but may (depending on how columns are defined in GLFSTMPLT3) include all values for that dim #.** |
| **Define a constant dimension for filtering** | **TEMPTMPLT1\_CONSTANTFILTERDIMNUM** | **Checkbox** | **-** | **NULL unless GLFSTMPLT1\_FILTERONDIMNUM is TRUE. If TRUE, GLFSTMPLT1\_CONSTANTDIMNUM is the # of the constant dimension. If FALSE, user will be prompted for dimension value at run time.** |
| **Define a constant dimension value for filtering** | **TEMPTMPLT1\_CONSTANTFILTERDIMVAL** | **Checkbox** | **-** | **NULL unless GLFSTMPLT1\_FILTERONDIMVAL is TRUE. If TRUE, GLFSTMPLT1\_CONSTANTDIMNUM is the # of the constant dimension and GLFSTMPLT1\_CONSTANTDIMVAL is the value. If FALSE, user will be prompted for dimension value at run time.** |
| **Filter report on this division** | **TEMPTMPLT1\_DIV** | **Lookup to GLDIV** | **-** | **Translate to new record ID when writing new GLFSTMPLT1.** |
| **Filter report on this project** | **TEMPTMPLT1\_PROJ** | **Lookup to GLPROJ** | **-** | **Translate to new record ID when writing new GLFSTMPLT1.** |
| **Filter report on this dimension** | **TEMPTMPLT1\_CONSTANTDIMNUM** | **Number(2,0)** | **-** | **NULL unless GLFSTMPLT1\_CONSTANTFILTERDIMNUM is TRUE.** |

*Table 13 –* ***TEMPTMPLT1***

Table **TEMPTMPLT2** – Temporary version of **GLFSTMPLT2** – variant header.

| **Field label** | **Column** | **Type/size** | **Possible values** | **Meaning / comment** |
| --- | --- | --- | --- | --- |
|  | **TEMPTMPLT2\_GLFSTMPLT1** | **Master/detail to GLFSTMPLT1** | **-** | **Translate to new record ID when writing new GLFSTMPLT2.** |
| **ID of this variant of the template** | **TEMPTMPLT2\_FORMAT** | **Text(30)** | **Not NULL** | **Identifies a variant of the template. Cannot be NULL. If no variants are defined, the GLFSTMPLT2\_FORMAT for the first one is defaulted to "Original version."** |
| **Number of columns on report** | **TEMPTMPLT2\_COLUMNS** | **Number(2,0)** | **> 0** | **Statements are columnar. The leftmost column of every statement (column zero) always contains text and never has a header; this column is assumed. GLFSTMPLT2\_COLUMNS refers to the number of data columns to be printed to the right of the text.** |
| **Number of tab-spaces to left of first column** | **TEMPTMPLT2\_COLZEROOFFSET** | **Number(2,0)** | **>= 0** | **Number of tab-spaces by which column zero is offset from the left edge of the page.** |
| **Width of leftmost (text) column in characters** | **TEMPTMPLT2\_COLZEROWIDTH** | **Number(2,0)** | **> 0** | **Width of column zero in characters. If more than GLFSTMPLT2\_COLZEROWIDTH characters are entered for any line in column zero, the text will wrap.** |
| **Maximum number of lines in any column heading** | **TEMPTMPLT2\_HEADERLINES** | **Number(1,0)** | **> 0 and <= 5** | **Number of column header lines (1, 2, 3, 4 or 5)** |
| **Currency precision** | **TEMPTMPLT2\_CURRPRECISION** | **Number(1,0)** | **0 or 2** | **The decimal precision of the reporting currency. Defaults to 2 but may be overridden to zero (as for Yen).** |
| **Currency rounding** | **TEMPTMPLT2\_ROUNDCURR** | **Picklist** | **Don't round** | **The default.** |
| **Round to even units** | **Round to even units of currency.** |
| **Round to thousands** | **Round to even thousands.** |
| **Round to millions** | **Round to even millions.** |
| **What to round** | **TEMPTMPLT2\_ROUNDWHAT** | **Picklist** | **NULL** | **If GLFSTMPLT2\_ROUNDCURR = Don't round** |
| **Lines only** | **Only round the numbers that appear on report lines. Add the unrounded line value to totals. When the value of the line is referenced in a calculation, use the unrounded value.** |
| **Lines and totals** | **Round the numbers that appear on report lines, and add the rounded values to totals. When the value of the line is referenced in a calculation, use the unrounded value.** |
| **Lines and calculation values** | **Round the numbers that appear on report lines. Add the unrounded line value to totals. When the value of the line is referenced in a calculation, use the rounded value.** |
| **Everything** | **Round the numbers that appear on report lines. Add the rounded line value to totals. When the value of the line is referenced in a calculation, use the rounded value.** |
| **How negative numbers are identified** | **TEMPTMPLT2\_MINUSSIGN** | **Picklist** | **Parentheses** | **The default. (999)** |
| **Leading dash** | **-999** |
| **Trailing dash** | **999-** |
| **Decimal point and thousands separator** | **TEMPTMPLT2\_PUNCTUATION** | **Picklist** | **Decimal = dot, 1000s = comma** | **Use dot for decimal point and comma for thousands separator: 999,999,999.99** |
| **Decimal = comma, 1000s = dot** | **Use comma for decimal point and dot for thousands separator: 999.999.999,99** |
| **Percentage / small number precision** | **TEMPTMPLT2\_PCTPRECISION** | **Number(1,0)** | **0, 1, or 2** | **Number of factional digits to print for percentages and other small numbers.** |
| **Character to print when a number column is empty** | **TEMPTMPLT2\_NULLVALUEINDICATOR** | **Text(1)** | **"-" or <blank>** | **Character (dash or blank) to display at the intersections of calculated lines and calculated columns. Regardless of the value of GLFSTMPLT2\_NULLVALUEINDICATOR, blanks are output when exporting reports.** |
| **Print a dash when a number column equals zero** | **TEMPTMPLT2\_SHOWDASHFORZEROVALUE** | **Checkbox** | **-** | **If GLFSTMPLT2\_SHOWDASHFORZEROVALUE is TRUE, a dash is output in the position immediately to the left of the decimal indicator when number columns equal zero; otherwise a formatted zero is output. Regardless of the value of GLFSTMPLT2\_ZEROVALUEINDICATOR, zeroes are output when exporting reports.** |
| **Number of centered title lines at the top of the page** | **TEMPTMPLT2\_TITLELINES** | **Number(2,0)** | **>=0 and < 11** | **Number of title lines at top of report (title lines are centered)** |
| **Top title line** | **TEMPTMPLT2\_TITLELINE1** | **Text(70)** | **-** | **NULL if GLFSTMPLT2\_TITLELINES = 0.  Predefined macros that can be embedded in GLFSTMPLT3\_TITKELINE1...10 and resolved at run time:  "CO" = Company # [GLCNTL\_CMPNO]  "COMPANY" = Company name [GLCNTL\_CMPNO\_\_R.SYCMP\_DESCR]  "PROJ" = Project number [GLPROJ\_PJPROJ]  "PROJECT" = Project title [GLPROJ\_PJPROJ\_\_R.PJPROJ\_DESCR]  "DIV" = Division code [GLDIV\_DIV]  "DIVISION" = Division name [GLDIV\_DESCR]  "MMMM" = January, February, etc., as of report date  "MMM" = Jan, Feb, etc., as of report date  "MM" = month of report as 01, 02, etc.  "YYYY" = year of report date as 2014, 2015, etc.  "YY" = year of report date as 14, 15, etc.  "DD" = day of report date as 1,2…30,31  "DIM#" = dimension # (1-8)  "DIMSRC" = dimension source name  "DIMVAL" = dimension value  "[nn]" (where nn is a number from 1 to 70) can be appended to any token (e.g. DIVISION[30]). If the value of token exceeds nn characters in length, only the leftmost nn characters are used.** |
| **Second title line** | **TEMPTMPLT2\_TITLELINE2** | **Text(70)** | **-** | **NULL if GLFSTMPLT2\_TITLELINES < 2** |
| **Third title line** | **TEMPTMPLT2\_TITLELINE3** | **Text(70)** | **-** | **NULL if GLFSTMPLT2\_TITLELINES < 3** |
| **Fourth title line** | **TEMPTMPLT2\_TITLELINE4** | **Text(70)** | **-** | **NULL if GLFSTMPLT2\_TITLELINES < 4** |
| **Fifth title line** | **TEMPTMPLT2\_TITLELINE5** | **Text(70)** | **-** | **NULL if GLFSTMPLT2\_TITLELINES < 5** |
| **Sixth title line** | **TEMPTMPLT2\_TITLELINE6** | **Text(70)** | **-** | **NULL if GLFSTMPLT2\_TITLELINES < 6** |
| **Seventh title line** | **TEMPTMPLT2\_TITLELINE7** | **Text(70)** | **-** | **NULL if GLFSTMPLT2\_TITLELINES < 7** |
| **Eighth title line** | **TEMPTMPLT2\_TITLELINE8** | **Text(70)** | **-** | **NULL if GLFSTMPLT2\_TITLELINES < 8** |
| **Ninth title line** | **TEMPTMPLT2\_TITLELINE9** | **Text(70)** | **-** | **NULL if GLFSTMPLT2\_TITLELINES < 9** |
| **Tenth title line** | **TEMPTMPLT2\_TITLELINE10** | **Text(70)** | **-** | **NULL if GLFSTMPLT2\_TITLELINES < 10** |
| **Unresolved line labels exist** | **TEMPTMPLT2\_UNRESOLVEDLABELS** | **Checkbox** | **-** | **No user input - set by trigger when format is saved.  If GLFSTMPLT2\_UNRESOLVEDLABELS is TRUE, no report can be crated from the format. If there are unresolved line labels - e.g. GLFSTMPLT3\_VALUE*n*CALCLABEL is not NULL but no corresponding GLFSTMPLT4\_CALCLABEL exists - the template is unusable, the user is warned, and GLFSTMPLT2\_UNRESOLVEDLABELS is set to TRUE - meaning that statement calculation cannot be run for GLFSTMPLT2\_FORMAT.** |
| **Looping calculations are defined** | **TEMPTMPLT2\_LOOPINCALCULATION** | **Checkbox** | **-** | **Not maintainable by user. Initialized to NULL. When any GLFSTMPLT3, GLFSTMPLT4, or GLFSTMPLT5 is saved:  GLFSTMPLT2\_LOOPINCALCULATION is set to FALSE if no looping calculations are defined.  GLFSTMPLT2\_LOOPINCALCULATION is set to TRUE if looping calculations exist. If GLFSTMPLT2\_LOOPINCALCULATION is TRUE when a user attempts to generate a report, the report generation aborts with an error message (e.g. "Cannot create statement - loop in calculations exists.")** |
|  |  |  |  | **The following columns permit overrides to GLFSTMPLT1 data. They are NULL unless GLFSTMPLT1-level data is being overwritten.** |
| **Override template-level report type** | **TEMPTMPLT2\_OVERRIDETYPE** | **Checkbox** | **-** | **If TRUE, override GLFSTMPLT1\_REPORTTYPE** |
| **Override template-level report scope** | **TEMPTMPLT2\_OVERRIDESCOPE** | **Checkbox** | **-** | **If TRUE, override GLFSTMPLT1\_SCOPE** |
| **Override template-level division filter** | **TEMPTMPLT2\_OVERRIDEDIVFILTER** | **Checkbox** | **-** | **If TRUE, override GLFSTMPLT1 division filters** |
| **Override template-level project filter** | **TEMPTMPLT2\_OVERRIDEPROJFILTER** | **Checkbox** | **-** | **If TRUE, override GLFSTMPLT1 project filters** |
| **Override template-level dimension filter** | **TEMPTMPLT2\_OVERRIDEDIMFILTER** | **Checkbox** | **-** | **If TRUE, override GLFSTMPLT1 dimension filters** |
| **Report type** | **TEMPTMPLT2\_REPORTTYPE** | **Picklist** | **NULL** | **If GLFSTMPLT2\_OVERRIDETYPE = FALSE** |
| **Balance sheet** | **Formal balance sheet** |
| **Balance sheet schedule** | **Supporting schedule to balance sheet** |
| **P & L** | **Formal P & L** |
| **P & L schedule** | **Supporting schedule to P & L** |
| **Cash flow statement** | **Cash flow statement (aka statement of changes to financial position). Not supported in initial release.** |
| **Cash flow stmt schedule** | **Supporting schedule to cash flow statement. Not supported in initial release.** |
| **Report scope** | **TEMPTMPLT2\_SCOPE** | **Picklist** | **NULL Company wide Single division Single project Company plus dimension Division plus dimension Project plus dimension** | **NULL if GLFSTMPLT2\_OVERRIDESCOPE = FALSE Must be Company wide if GLFSTMPLT2\_REPORTTYPE = Balance sheet. Must be Company wide or Company plus dimension if GLFSTMPLT2\_REPORTTYPE = Balance sheet schedule.** |
| **Consolidated or Single-Company Report?** | **TEMPTMPLT2\_ONECOMPORCONSOL** | **Picklist** | **Single Company**  **Consolidated** | **Default to Single Company.**  **If Single Company, GLFSTMPLT5\_CMPNO must be Reporting Company.**  **If Consolidated, GLFSTMPLT5\_CMPNO may be any company in the chain of consolidation.** |
| **Define an constant division for filtering** | **TEMPTMPLT2\_CONSTANTDIV** | **Checkbox** | **-** | **NULL unless GLFSTMPLT2\_SCOPE = Single division, Single project, Division plus dimension, or Project plus dimension. If TRUE, this template only applies to GLFSTMPLT2\_DIV. If FALSE, user will be prompted for division at run time.** |
| **Define an constant project for filtering** | **TEMPTMPLT2\_CONSTANTPROJ** | **Checkbox** | **-** | **NULL unless GLFSTMPLT2\_SCOPE = Single project or Project plus dimension. NULL if GLFSTMPLT2\_REPORTTYPE = Balance sheet. If TRUE, this template only applies to GLFSTMPLT2\_PROJ. If FALSE, user will be prompted for project at run time.** |
| **Filter report on this division** | **TEMPTMPLT2\_DIV** | **Lookup to GLDIV** | **-** | **Translate to new record ID when writing new GLFSTMPLT2.** |
| **Filter report on this project** | **TEMPTMPLT2\_PROJ** | **Lookup to GLPROJ** | **-** | **Translate to new record ID when writing new GLFSTMPLT2.** |
| **Filter whole report on a single dimension** | **TEMPTMPLT2\_FILTERONDIMNUM** | **Checkbox** | **-** | **NULL unless GLFSTMPLT2\_SCOPE = Company plus dimensions or Single div plus dimensions or Single proj plus dimensions. If TRUE, the whole report will be confined to a single dim #. If FALSE, use of dimensions (if any) is confined to column definitions (in GLFSTMPLT3).** |
| **Filter whole report on a value for a single dimension** | **TEMPTMPLT2\_FILTERONDIMVAL** | **Checkbox** | **-** | **NULL unless GLFSTMPLT2\_FILTERONDIMNUM is TRUE. If TRUE, the whole report will be confined to a single pair of dim # and dim value. If FALSE, the report will be confined to a single dim # but may (depending on how columns are defined in GLFSTMPLT3) include all values for that dim #.** |
| **Define an constant dimension for filtering** | **TEMPTMPLT2\_CONSTANTFILTERDIMNUM** | **Checkbox** | **-** | **NULL unless GLFSTMPLT2\_FILTERONDIMNUM is TRUE. If TRUE, GLFSTMPLT2\_CONSTANTFILTERDIMNUM is the # of the constant dimension. If FALSE, user will be prompted for dimension value at run time.** |
| **Define a constant dimension value for filtering** | **TEMPTMPLT2\_CONSTANTFILTERDIMVAL** | **Checkbox** | **-** | **NULL unless GLFSTMPLT2\_FILTERONDIMVAL is TRUE. If TRUE, GLFSTMPLT2\_CONSTANTFILTERDIMNUM is the # of the constant dimension and GLFSTMPLT2\_CONSTANTFILTERDIMVAL is the value. If FALSE, user will be prompted for dimension value at run time.** |
| **Filter report on this dimension** | **TEMPTMPLT2\_CONSTANTDIMNUM** | **Number(2,0)** | **-** | **NULL unless GLFSTMPLT2\_CONSTANTFILTERDIMNUM is TRUE.** |
| **Filter report on this dimension value** | **TEMPTMPLT2\_CONSTANTDIMVAL** | **Text(20)** | **-** | **NULL unless GLFSTMPLT2\_CONSTANTFILTERDIMVAL is TRUE.** |
|  |  |  |  | **The following columns are set in Process 9** |
| **Last report number created** | **TEMPTMPLT2\_LASTREPORT** | **Loolup to GLFINSTMT1** | **-** | **Translate as null.** |
| **The template is in error** | **TEMPTMPLT2\_INERROR** | **Checkbox** | **-** | **The statement cannot be created.** |
| **There is a loop in the calculations** | **TEMPTMPLT2\_LOOPINGCALCULATION** | **Checkbox** | **-** | **There is a loop in the calculation – a calculation depends on its own value.** |
| **There is an infinite regression** | **TEMPTMPLT2\_INFINITEREGRESS** | **Checkbox** | **-** | **A total includes a calculation line that references a total that includes a calculation line that references a total – and so on, infinitely.** |
| **The template is incomplete** | **TEMPTMPLT2\_INCOMPLETETEMPLATE** | **Checkbox** | **-** | **Lines are referenced that don’t exist ot columns are not defined.** |

*Table 14 –* ***TEMPTMPLT2***

Table **TEMPTMPLT3** – Temporary version of **GLFSTMPLT3** – column definitions.

| **Field label** | **Column** | **Type/size** | **Possible values** | **Meaning / comment** |
| --- | --- | --- | --- | --- |
|  | **TEMPTMPLT3\_GLFSTMPLT2** | **Master/detail to GLFSTMPLT2** | **-** | **Translate to new record ID when writing new GLFSTMPLT3.** |
| **Column number** | **TEMPTMPLT3\_COLNO** | **Number(2,0)** | **> 0 and <= GLFSTMPLT2\_COLUMNS** |  |
| **Non-printing column flag** | **TEMPTMPLT3\_NONPRINTING** | **Checkbox** | **-** |  |
| **Print column values as percentages or other small numbers** | **TEMPTMPLT3\_PRINTASPERCENTAGE** | **Checkbox** | **-** |  |
| **Top line of column heading** | **TEMPTMPLT3\_HEADERLINE1** | **Text(30)** | **-** |  |
| **Second line of column heading** | **TEMPTMPLT3\_HEADERLINE2** | **Text(30)** | **-** |  |
| **Third line of column heading** | **TEMPTMPLT3\_HEADERLINE3** | **Text(30)** | **-** |  |
| **Fourth line of column heading** | **TEMPTMPLT3\_HEADERLINE4** | **Text(30)** | **-** |  |
| **Fifth line of column heading** | **TEMPTMPLT3\_HEADERLINE5** | **Text(30)** | **-** |  |
| **Type and origin of column values** | **TEMPTMPLT3\_VALUETYPE** | **Picklist** | **GL** |  |
| **Statement entry** |  |
| **GL + Statement entry** |  |
| **Budget** |  |
| **Calculated** |  |
| **Status for identifying budget data** | **TEMPTMPLT3\_BUDSTATUS** | **Picklist** | **Official base budget Official revision** |  |
| **How revision level for identifying budget data is known** | **TEMPTMPLT3\_BUDREVLEVELMETHOD** | **Picklist** | **Constant Variable** |  |
| **Constant revision level** for identifying budget data | **TEMPTMPLT3\_BUDREVLEVEL** | **Text(10)** | **-** |  |
| **How effective year of column is known** | **TEMPTMPLT3\_CURRENTYEARMEANS** | **Picklist** | **Floating Absolute Entered at run time**  **Calculated column**  **Floating period with offset**  **Floating year with offset** |  |
| **How effective period of column is known** | **TEMPTMPLT3\_CURRENTPERIODMEANS** | **Picklist** | **Floating Absolute Entered at run time Calculated column**  **Floating period with offset** |  |
| **Constant effective year of column** | **TEMPTMPLT3\_FIXEDYEAR** | **Number(4,0)** | **-** |  |
| **Constant effective period of column** | **TEMPTMPLT3\_FIXEDPERIODNO** | **Number(2,0)** | **NULL or ( > 0 and < 14)** |  |
| **Offset from reporting period** | **TEMPTMPLT3\_PERIODOFFSET** | **Number(2,0)** | **> -100 and < 100** |  |
| **Offset from reporting year** | **TEMPTMPLT3\_YEAROFFSET** | **Number(1,0)** | **> -10 and < 10** |  |
| **How to treat period zero if period is offset** | **TEMPTMPLT3\_PERIODZEROPOLICY** | **Picklist** | **Skip**  **Add to period 1**  **Show by itself** |  |
| **How to treat period 13 if period is offset** | **TEMPTMPLT3\_PERIOD13POLICY** | **Picklist** | **Skip**  **Add to period 12**  **Show by itself** |  |
| **Timeframe of column** | **TEMPTMPLT3\_PTDYTD** | **Picklist** | **Period totals Year to date totals YTD period average Calculated column** |  |
| **Label of external data exported from column** | **TEMPTMPLT3\_EXPORTLABEL** | **Text(50)** | **-** |  |
| **Filter on division** | **TEMPTMPLT3\_FILTERONDIV** | **Checkbox** |  |  |
| **Filter on project** | **TEMPTMPLT3\_FILTERONPROJ** | **Checkbox** |  |  |
| **Filter on dimension** | **TEMPTMPLT3\_FILTERONDIM** | **Checkbox** |  |  |
| **Division filtering method** | **TEMPTMPLT3\_DIVMETHOD** | **Picklist** | **NULL** |  |
| **All divisions** |  |
| **Variable division** |  |
| **Constant division** |  |
| **No division** |  |
| **Constant division for filtering** | **TEMPTMPLT3\_DIV** | **Lookup to GLDIV** |  | **Translate to new record ID when writing new GLFSTMPLT3.** |
| **Project filtering method** | **TEMPTMPLT3\_PROJMETHOD** | **Picklist** | **NULL** |  |
| **All projects** |  |
| **Variable project** |  |
| **Constant project** |  |
| **No project** |  |
| **Constant project for filtering** | **TEMPTMPLT3\_PROJ** | **Lookup to GLPROJ** |  | **Translate to new record ID when writing new GLFSTMPLT3.** |
| **Dimension filtering method** | **TEMPTMPLT3\_DIMMETHOD** | **Picklist** | **NULL** |  |
| **All dimensions** |  |
| **Constant dimension, constant value** |  |
| **Constant dimension, exclude constant value** |  |
| **Constant dimension, variable value** |  |
| **Constant dimension, exclude variable value** |  |
| **Constant dimension, all values** |  |
| **Variable dimension, variable value** |  |
| **Variable dimension, exclude variable value** |  |
| **Variable dimension, all values** |  |
| **All dimensions, all values** |  |
| **No dimension** |  |
| **Constant dimension for filtering** | **TEMPTMPLT3\_DIMNUM** | **Number(2,0)** | **Dimension #** |  |
| **Constant dimension value for filtering** | **TEMPTMPLT3\_DIMVAL** | **Text(20)** | **Dimension value** |  |
| **Expression operator for calculation** | **TEMPTMPLT3\_OPERATOR** | **Picklist** | **Plus Minus Multiplied by Divided by As a percentage of No operator** |  |
| **Origin of equation left side** | **TEMPTMPLT3\_VALUE1TYPE** | **Picklist** | **Same row, other column Other row, other column Constant Entered at runtime Imported at runtime Special** |  |
| **Origin of equation right side** | **TEMPTMPLT3\_VALUE2TYPE** | **Picklist** | **Same row, other column Other row, other column Constant Entered at runtime Imported at runtime Special** |  |
| **Constant value of equation left side** | **TEMPTMPLT3\_VALUE1CONSTANT** | **Number(16,2)** | **-** |  |
| **Constant value of equation right side** | **TEMPTMPLT3\_VALUE2CONSTANT** | **Number(16,2)** | **-** |  |
| **Type of special value for equation left side** | **TEMPTMPLT3\_VALUE1SPECIALTYPE** | **Picklist** | **Period number Periods in year** |  |
| **Type of special value for equation right side** | **TEMPTMPLT3\_VALUE2SPECIALTYPE** | **Picklist** | **Period number Periods in year** |  |
| **Line label for source of equation left side** | **TEMPTMPLT3\_VALUE1CALCLABEL** | **Text(50)** | **-** |  |
| **Line label for source of equation right side** | **TEMPTMPLT3\_VALUE2CALCLABEL** | **Text(50)** | **-** |  |
| **Starting column for source of equation left side** | **TEMPTMPLT3\_VALUE1COLFROM** | **Number(2,0)** | **-** |  |
| **Ending column for source of equation left side** | **TEMPTMPLT3\_VALUE1COLTHRU** | **Number(2,0)** | **-** |  |
| **Starting column for source of equation right side** | **TEMPTMPLT3\_VALUE2COLFROM** | **Number(2,0)** | **-** |  |
| **Ending column for source of equation right side** | **TEMPTMPLT3\_VALUE2COLTHRU** | **Number(2,0)** | **-** |  |
| **Negate results of expression** | **TEMPTMPLT3\_NEGATEEXPRESSION** | **Checkbox** | **-** |  |
| **Total policy for column** | **TEMPTMPLT3\_TOTALPOLICY** | **Picklist** | **Add to totals** |  |
| **Calculate total lines** |  |
| **Null total lines** |  |
| **Add to totals, then null line** |  |
| **Calculate total lines, null other lines** |  |

*Table 15 –* ***TEMPTMPLT3***

Table **TEMPTMPLT4** – Temporary version of **GLFSTMPLT4** – statement line header.

| **Field label** | **Column** | **Type/size** | ***Possible values*** | **Meaning / comment** |
| --- | --- | --- | --- | --- |
|  | **TEMPTMPLT4\_GLFSTMPLT2** | **Master/detail to GLFSTMPLT2** | ***-*** | **Translate to new record ID when writing new GLFSTMPLT4.** |
| **Line number** | **TEMPTMPLT4\_LINENO** | **Number(4,0)** | ***-*** |  |
| **Non-printing line flag** | **TEMPTMPLT4\_NONPRINTING** | **Checkbox** | ***-*** |  |
| **Do not add line results to totals** | **TEMPTMPLT4\_NOTOTAL** | **Checkbox** | ***-*** |  |
| **Line command** | **TEMPTMPLT4\_COMMAND** | **Picklist** | ***-*** |  |
| ***Show debits as positive*** |  |
| ***Show credits as positive*** |  |
| ***Change active total group*** |  |
| ***Do a page break*** |  |
| ***Print a blank line*** |  |
| ***Print a single underline*** |  |
| ***Print a double underline*** |  |
| ***Define a line name*** |  |
| ***Total level 1*** |  |
| ***Total level 2*** |  |
| ***Total level 3*** |  |
| ***Total level 4*** |  |
| ***Total level 5*** |  |
| ***Total level 6*** |  |
| ***Total level 7*** |  |
| ***Total level 8*** |  |
| ***Total level 9*** |  |
| ***Accumulation line*** |  |
| ***Calculation line*** |  |
| **New total group** | **TEMPTMPLT4\_NEWTOTALGROUP** | **Number(1,0)** | ***1 - 9*** |  |
| **Printed line text** | **TEMPTMPLT4\_TEXT** | **Text(50)** | ***-*** |  |
| **Indent text this many tabs** | **TEMPTMPLT4\_TEXTINDENT** | **Number(1,0)** | ***0, 1, 2, or 3*** |  |
| **Bold the text** | **TEMPTMPLT4\_TEXTBOLD** | **Checkbox** |  |  |
| **Italicize the text** | **TEMPTMPLT4\_TEXTITALICS** | **Checkbox** |  |  |
| **Underline the text** | **TEMPTMPLT4\_TEXTUNDERLINE** | **Checkbox** |  |  |
| **Print line results as percentages** | **TEMPTMPLT4\_PRINTASPERCENTAGE** | **Checkbox** | ***-*** |  |
| **Line label for reference by calculations** | **TEMPTMPLT4\_CALCLABEL** | **Text(50)** | ***-*** |  |
| **Filter on division** | **TEMPTMPLT4\_FILTERONDIV** | **Checkbox** |  |  |
| **Filter on project** | **TEMPTMPLT4\_FILTERONPROJ** | **Checkbox** |  |  |
| **Filter on dimension** | **TEMPTMPLT4\_FILTERONDIM** | **Checkbox** |  |  |
| **Division filtering method** | **TEMPTMPLT4\_DIVMETHOD** | **Picklist** | ***NULL*** |  |
| ***Variable division*** |  |
| ***Constant division*** |  |
| ***No division*** |  |
| **Constant division for filtering** | **TEMPTMPLT4\_DIV** | **Lookup to GLDIV** |  | **Translate to new record ID when writing new GLFSTMPLT4.** |
| **Project filtering method** | **TEMPTMPLT4\_PROJMETHOD** | **Picklist** | ***NULL*** |  |
| ***Variable project*** |  |
| ***Constant project*** |  |
| ***No project*** |  |
| **Constant project for filtering** | **TEMPTMPLT4\_PROJ** | **Lookup to GLPROJ** |  | **Translate to new record ID when writing new GLFSTMPLT4.** |
| **Dimension filtering method** | **TEMPTMPLT4\_DIMMETHOD** | **Picklist** | ***No dimension*** |  |
| ***Constant dimension, constant value*** |  |
| ***Constant dimension, exclude constant value*** |  |
| ***Constant dimension, variable value*** |  |
| ***Constant dimension, exclude variable value*** |  |
| ***Constant dimension, all values*** |  |
| ***Variable dimension, variable value*** |  |
| ***Variable dimension, exclude variable value*** |  |
| ***Variable dimension, all values*** |  |
| ***All dimensions, all values*** |  |
| ***No dimension*** |  |
| **Constant dimension for filtering** | **TEMPTMPLT4\_DIMNUM** | **Number(2,0)** | ***Dimension #*** |  |
| **Constant dimension value for filtering** | **TEMPTMPLT4\_DIMVAL** | **Text(20)** | ***Dimension value*** |  |
| **Expression operator** | **TEMPTMPLT4\_OPERATOR** | **Picklist** | ***Plus Minus Multiplied by Divided by As a percentage of No operator*** |  |
| **Origin of equation left side** | **TEMPTMPLT4\_VALUE1TYPE** | **Picklist** | ***Other row, same column Other row, other column Constant Entered at runtime Imported at runtime Special*** |  |
| **Origin of equation right side** | **TEMPTMPLT4\_VALUE2TYPE** | **Picklist** | ***Other row, same column Other row, other column Constant Entered at runtime Imported at runtime Special*** |  |
| **Constant value of equation left side** | **TEMPTMPLT4\_VALUE1CONSTANT** | **Number(16,2)** | ***-*** |  |
| **Constant value of equation right side** | **TEMPTMPLT4\_VALUE2CONSTANT** | **Number(16,2)** | ***-*** |  |
| **Type of special value for equation left side** | **TEMPTMPLT4\_VALUE1SPECIALTYPE** | **Picklist** | ***Period number Periods in year*** |  |
| **Type of special value for equation right side** | **TEMPTMPLT4\_VALUE2SPECIALTYPE** | **Picklist** | ***Period number Periods in year*** |  |
| **Line label for source of equation left side** | **TEMPTMPLT4\_VALUE1CALCLABEL** | **Text(50)** | ***-*** |  |
| **Line label for source of equation right side** | **TEMPTMPLT4\_VALUE2CALCLABEL** | **Text(50)** | ***-*** |  |
| **Column for source of equation left side** | **TEMPTMPLT4\_VALUE1COL** | **Number(2,0)** | ***-*** |  |
| **Column for source of equation right side** | **TEMPTMPLT4\_VALUE2COL** | **Number(2,0)** | ***-*** |  |
| **Negate results of expression** | **TEMPTMPLT4\_NEGATEEXPRESSION** | **Checkbox** | ***-*** |  |
| **First label for external data exported from this line** | **TEMPTMPLT4\_EXPORTLABEL1** | **Text(50)** | ***-*** |  |
| **Second label for external data exported from this line** | **TEMPTMPLT4\_EXPORTLABEL2** | **Text(50)** | ***-*** |  |
| **Third label for external data exported from this line** | **TEMPTMPLT4\_EXPORTLABEL3** | **Text(50)** | ***-*** |  |
| **Fourth label for external data exported from this line** | **TEMPTMPLT4\_EXPORTLABEL4** | **Text(50)** | ***-*** |  |
| **Fifth label for external data exported from this line** | **TEMPTMPLT4\_EXPORTLABEL5** | **Text(50)** | ***-*** |  |
| **Sixth label for external data exported from this line** | **TEMPTMPLT4\_EXPORTLABEL6** | **Text(50)** | ***-*** |  |
| **Seventh label for external data exported from this line** | **TEMPTMPLT4\_EXPORTLABEL7** | **Text(50)** | ***-*** |  |
| **Eighth label for external data exported from this line** | **TEMPTMPLT4\_EXPORTLABEL8** | **Text(50)** | ***-*** |  |
| **First label to be matched to a GLFSTMPLT3\_EXPORTLABEL** | **TEMPTMPLT4\_COLEXPORTLABEL1** | **Text(50)** | ***-*** |  |
| **Second label to be matched to a GLFSTMPLT3\_EXPORTLABEL** | **TEMPTMPLT4\_COLEXPORTLABEL2** | **Text(50)** | ***-*** |  |
| **Third label to be matched to a GLFSTMPLT3\_EXPORTLABEL** | **TEMPTMPLT4\_COLEXPORTLABEL3** | **Text(50)** | ***-*** |  |
| **Fourth label to be matched to a GLFSTMPLT3\_EXPORTLABEL** | **TEMPTMPLT4\_COLEXPORTLABEL4** | **Text(50)** | ***-*** |  |
| **Fifth label to be matched to a GLFSTMPLT3\_EXPORTLABEL** | **TEMPTMPLT4\_COLEXPORTLABEL5** | **Text(50)** | ***-*** |  |
| **Sixth label to be matched to a GLFSTMPLT3\_EXPORTLABEL** | **TEMPTMPLT4\_COLEXPORTLABEL6** | **Text(50)** | ***-*** |  |
| **Seventh label to be matched to a GLFSTMPLT3\_EXPORTLABEL** | **TEMPTMPLT4\_COLEXPORTLABEL7** | **Text(50)** | ***-*** |  |
| **Eighth label to be matched to a GLFSTMPLT3\_EXPORTLABEL** | **TEMPTMPLT4\_COLEXPORTLABEL8** | **Text(50)** | ***-*** |  |

*Table 16 –* ***TEMPTMPLT4***

Table **TEMPTMPLT5** – Temporary version of **GLFSTMPLT5** – statement line calculation detail.

| **Field label** | **Column** | **Type/size** | **Possible values** | **Meaning / comment** |
| --- | --- | --- | --- | --- |
|  | **TEMPTMPLT5\_GLFSTMPLT4** | **Master/detail to GLFSTMPLT4** | **-** | **Translate to new record ID when writing new GLFSTMPLT5.** |
| **Subline sequence #** | **TEMPTMPLT5\_LINESEQ** | **Number(4,0)** | **> 0** |  |
| **Subline command** | **TEMPTMPLT5\_COMMAND** | **Picklist** | **Comment** |  |
| **Add an account** |  |
| **Subtract an account** |  |
| **Add an account range** |  |
| **Subtract an account range** |  |
| **Invoke a predefined subline group** |  |
| **Add labeled line** |  |
| **Subtract labeled line** |  |
| **Multiply by labeled line** |  |
| **Divide by labeled line** |  |
| **Add imported data** |  |
| **Subtract imported data** |  |
| **Multiply by imported data** |  |
| **Divide by imported data** |  |
| **Start of account range** | **TEMPTMPLT5\_RANGESTART** | **Text(30)** | **-** |  |
| **End of account range** | **TEMPTMPLT5\_RANGEEND** | **Text(30)** | **>= GLFSTMPLT5\_RANGESTART** |  |
| **Company for accounts** | **TEMPTMPLT5\_CMPNO** | **Lookup to GLCMP** | **Must be the reporting company or a subsidiary of the reporting company** | **Translate to new record ID when writing new GLFSTMPLT5.** |
| **Account number** | **TEMPTMPLT5\_GLACCT** | **Lookup to GLACCT** | **-** | **Translate to new record ID when writing new GLFSTMPLT5.** |
| **Label of line that is data source** | **TEMPTMPLT5\_SOURCELINE** | **Text(50)** | **-** |  |
| **Invoke this group of predefined sublines** | **TEMPTMPLT5\_GLFSTMPLT7** | **Lookup to GLFSTMPLT7** | **-** | **Translate to new record ID when writing new GLFSTMPLT5.** |
| **Comment** | **TEMPTMPLT5\_TEXT** | **Text(100)** | **-** |  |

*Table 17 –* ***TEMPTMPLT5***

Table **TEMPTMPLT6** – Temporary version of **GLFSTMPLT6** – external data.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field label** | **Column** | **Type/size** | **Possible values** | **Meaning / comment** |
| **Company** | **TEMPTMPLT6\_CMPNO** | **Lookup to GLCMP** |  | **Translate to new record ID when writing new GLFSTMPLT6.** |
| **External data label** | **TEMPTMPLT6\_LABEL** | **Text(50)** | **Not NULL** |  |
| **Start of effective year range** | **TEMPTMPLT6\_FROMYEAR** | **Lookup to GLYEAR** |  | **Translate to new record ID when writing new GLFSTMPLT6.** |
| **Start of effective period range** | **TEMPTMPLT6\_FROMPERIOD** | **Number(2,0)** |  |  |
| **End of effective year range** | **TEMPTMPLT6\_THRUYEAR** | **Lookup to GLYEAR** |  | **Translate to new record ID when writing new GLFSTMPLT6.** |
| **End of effective period range** | **TEMPTMPLT6\_THRUPERIOD** | **Number(2,0)** |  |  |
| **Scale of decimal precision** | **TEMPTMPLT6\_SCALE** | **Picklist** | **Currency General** |  |
| **Currency amount** | **TEMPTMPLT6\_CURRENCYVAL** | **Number(16,2)** |  |  |
| **Non-currency amount** | **TEMPTMPLT6\_GENERALVAL** | **Number(10,8)** |  |  |

*Table 18 –* ***TEMPTMPLT6***

Table **TEMPTMPLT7** – Temporary version of **GLFSTMPLT7** – predefined subline group header.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field label** | **Column** | **Type/size** | **Possible values** | **Meaning / comment** |
| **Company** | **TEMPTMPLT7\_GLCMP** | **Lookup to GLCMP** | **-** | **Translate to new record ID when writing new GLFSTMPLT7.** |
| **Predefined subline group** | **TEMPTMPLT7\_NAME** | **Text(50)** | **Not NULL** |  |

*Table 19 –* ***TEMPTMPLT7***

Table **TEMPTMPLT8** – Temporary version of **GLFSTMPLT8** – predefined subline group detail.

| **Field label** | **Column** | **Type/size** | **Possible values** | **Meaning / comment** |
| --- | --- | --- | --- | --- |
|  | **TEMPTMPLT8\_GLFSTMPLT7** | **Master/detail to GLFSTMPLT7** | **-** | **Translate to new record ID when writing new GLFSTMPLT8.** |
| **Subline sequence #** | **TEMPTMPLT8\_SEQ** | **Number(4,0)** | **-** |  |
| **Subline command** | **TEMPTMPLT8\_COMMAND** | **Picklist** | **Add an account** |  |
| **Subtract an account** |  |
| **Add an account range** |  |
| **Subtract an account range** |  |
| **Start of account range** | **TEMPTMPLT8\_RANGESTART** | **Text(30)** | **-** |  |
| **End of account range** | **TEMPTMPLT8\_RANGEEND** | **Text(30)** | **>= GLFSTMPLT8\_RANGESTART** |  |
| **Company for accounts** | **TEMPTMPLT8\_CMPNO** | **Lookup to GLCMP** | **Must be the reporting company (GLFSTMPLT7\_GLCMP) or a subsidiary of the reporting company** | **Translate to new record ID when writing new GLFSTMPLT8.** |
| **Account number** | **TEMPTMPLT8\_GLACCT** | **Lookup to GLACCT** | **-** | **Translate to new record ID when writing new GLFSTMPLT8.** |

*Table 20 –* ***TEMPTMPLT8***

Table **TEMPTMPLT9** – Temporary version of **GLFSTMPLT9** – export label list for enforcing uniqueness.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field label** | **Column** | **Type/size** | **Possible values** | **Meaning / comment** |
| **Company** | **TEMPTMPLT9\_GLCMP** | **Lookup to GLCMP** | **-** | **Translate to new record ID when writing new GLFSTMPLT9.** |
| **External data label** | **TEMPTMPLT9\_EXPORTLABEL** | **Text(50)** | **-** |  |
| **Used in line** | **TEMPTMPLT9\_GLFSTMPLT4** | **Lookup to GLFSTMPLT4** | **-** | **Translate to new record ID when writing new GLFSTMPLT9.** |

*Table 21 –* ***TEMPTMPLT9***

**Process Spec**

**Screens**

|  |  |
| --- | --- |
| **Org-to-Org Financial Report Template Copy - Outbound Side** | |
|  |  |
| **Copy 1 template or all for 1 company?:** | **TMPLTMVR\_ONETMPLTORALL** |
| **Copy sublines?:** | **TMPLTMVR\_SUBLINES** |
| **Source GLCMP record name:** | **TMPLTMVR\_GLCMPNSRCNAME** |
| **Source template header record name:** | **TMPLTMVR\_GLFSTMPLT1SRCNAME** |
| **[Execute]** | **[Quit]** |

***Screen 1 –*** *Outbound side from Program 1. The words in brackets are buttons.*

|  |  |
| --- | --- |
| **Org-to-Org Financial Report Template Copy - Inbound Side** | |
|  |  |
|  | **Message area** |
|  |
|  |
|  |  |
|  |  |

***Screen 2 –*** *Inbound side from Program 2.*

**Record ID Translations**

Table 22, below, shows the record ID fields that must be translated from the source org to the destination org:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Table name (TMPLTMVRID\_TABLE)** | **TEMPTMPLT*n* Field Name (TMPLTMVRID\_FIELDNAME)** | **Points to Table** | **GLFSTMPLT*n* Field Name** | **Table**  **Processing Sequence** | **Encoding**  **Instruction # (Below)** | **Decoding**  **Instruction #s (Below)** |
| **GLFSTMPLT1** | **TEMPTMPLT1\_CMPNO** | **GLCMP** | **GLFSTMPLT1\_CMPNO** | **4** | **8** | **0.c, 4** |
| **GLFSTMPLT1** | **TEMPTMPLT1\_DIV** | **GLDIV** | **GLFSTMPLT1\_DIV** | **4** | **9** | **0.e, 4** |
| **GLFSTMPLT1** | **TEMPTMPLT1\_PROJ** | **GLPROJ** | **GLFSTMPLT1\_PROJ** | **4** | **10** | **0.f, 4** |
| **GLFSTMPLT2** | **TEMPTMPLT2\_GLFSTMPLT1** | **GLFSTMPLT1** | **GLFSTMPLT2\_GLFSTMPLT1** | **5** | **11** | **4.c, 5** |
| **GLFSTMPLT2** | **TEMPTMPLT2\_DIV** | **GLDIV** | **GLFSTMPLT2\_DIV** | **5** | **12** | **0.e, 5** |
| **GLFSTMPLT2** | **TEMPTMPLT2\_PROJ** | **GLPROJ** | **GLFSTMPLT2\_PROJ** | **5** | **13** | **0.f, 5** |
| **GLFSTMPLT3** | **TEMPTMPLT3\_GLFSTMPLT2** | **GLFSTMPLT2** | **GLFSTMPLT3\_GLFSTMPLT2** | **6** | **14** | **5.c, 6** |
| **GLFSTMPLT3** | **TEMPTMPLT3\_DIV** | **GLDIV** | **GLFSTMPLT3\_DIV** | **6** | **15** | **0.e, 6** |
| **GLFSTMPLT3** | **TEMPTMPLT3\_PROJ** | **GLPROJ** | **GLFSTMPLT3\_PROJ** | **6** | **16** | **0.f, 6** |
| **GLFSTMPLT4** | **TEMPTMPLT4\_GLFSTMPLT2** | **GLFSTMPLT2** | **GLFSTMPLT4\_GLFSTMPLT2** | **7** | **17** | **5.c, 7** |
| **GLFSTMPLT4** | **TEMPTMPLT4\_DIV** | **GLDIV** | **GLFSTMPLT4\_DIV** | **7** | **18** | **0.e, 7** |
| **GLFSTMPLT4** | **TEMPTMPLT4\_PROJ** | **GLPROJ** | **GLFSTMPLT4\_PROJ** | **7** | **19** | **0.f, 7** |
| **GLFSTMPLT5** | **TEMPTMPLT5\_GLFSTMPLT4** | **GLFSTMPLT4** | **GLFSTMPLT5\_GLFSTMPLT4** | **9** | **22** | **7.c, 9** |
| **GLFSTMPLT5** | **TEMPTMPLT5\_CMPNO** | **GLCMP** | **GLFSTMPLT5\_CMPNO** | **9** | **23** | **0.c, 9** |
| **GLFSTMPLT5** | **TEMPTMPLT5\_GLACCT** | **GLACCT** | **GLFSTMPLT5\_GLACCT** | **9** | **24** | **0.g, 9** |
| **GLFSTMPLT5** | **TEMPTMPLT5\_GLFSTMPLT7** | **GLFSTMPLT7** | **GLFSTMPLT5\_GLFSTMPLT7** | **9** | **25** | **2.c, 9** |
| **GLFSTMPLT6** | **TEMPTMPLT6\_CMPNO** | **GLCMP** | **GLFSTMPLT6\_CMPNO** | **1** | **1** | **0.c, 1** |
| **GLFSTMPLT6** | **TEMPTMPLT6\_FROMYEAR** | **GLYEAR** | **GLFSTMPLT6\_FROMYEAR** | **1** | **2** | **0.d, 1** |
| **GLFSTMPLT6** | **TEMPTMPLT6\_THRUYEAR** | **GLYEAR** | **GLFSTMPLT6\_THRUYEAR** | **1** | **3** | **0.d, 1** |
| **GLFSTMPLT7** | **TEMPTMPLT7\_GLCMP** | **GLCMP** | **GLFSTMPLT7\_GLCMP** | **2** | **4** | **0.c, 2** |
| **GLFSTMPLT8** | **TEMPTMPLT8\_GLFSTMPLT7** | **GLFSTMPLT7** | **GLFSTMPLT8\_GLFSTMPLT7** | **3** | **5** | **2.c, 3** |
| **GLFSTMPLT8** | **TEMPTMPLT8\_CMPNO** | **GLCMP** | **GLFSTMPLT8\_CMPNO** | **3** | **6** | **0.c, 3** |
| **GLFSTMPLT8** | **TEMPTMPLT8\_GLACCT** | **GLACCT** | **GLFSTMPLT8\_GLACCT** | **3** | **7** | **0.g, 3** |
| **GLFSTMPLT9** | **TEMPTMPLT9\_GLCMP** | **GLCMP** | **GLFSTMPLT9\_GLCMP** | **8** | **20** | **0.c, 8** |
| **GLFSTMPLT9** | **TEMPTMPLT9\_GLFSTMPLT4** | **GLFSTMPLT4** | **GLFSTMPLT9\_GLFSTMPLT4** | **8** | **21** | **7.c, 8** |

*Table 22 – The record ID fields to be convert from the values in the source org to the equivalent values in the destination org.*

Table 22 contains many of the detailed specifications for moving templates from the source org to the destination org. It will be necessary for the developer to refer constantly to Table 22, because the instructions given in this section will not be repeated in the pseudo-code.

Table 22, column by column:

Column A, **Table name (TMPLTMVRID\_TABLE):**

The name of the template specification table (**GLFSTMPLT1** through **GLFSTMPLT9**) to which the table row refers.

Column B, **TEMPTMPLT*n* Field Name (TMPLTMVRID\_FIELDNAME):**

The field name in the temporary equivalent of the table name in column A to which the row refers.

Column C, **Points to Table:**

The field named in columns B and D is a record ID in this table. The field named in column B holds the source org value of this record ID. The field named in column D will hold the destination org value of this field. The whole point of this process is to replace the source org values of these fields with the corresponding destination org values.

Column D, **GLFSTMPLT*n* Field Name:**

The field in the template specification table (**GLFSTMPLT1** through **GLFSTMPLT9**) in the destination org in which the converted record ID will go.

Column E, **Processing Sequence:**

The order in which the template specification tables are converted is very important because (obviously) we can’t get the record IDs of records that haven’t been created yet. Tables will be converted in this order:

1. **TEMPTMPLT6** to **GLFSTMPLT6** – Financial statement template table # 6 – external data.
2. **TEMPTMPLT7** to **GLFSTMPLT7** – Financial statement template table # 7 – predefined subline group header.
3. **TEMPTMPLT8** to **GLFSTMPLT8** – Financial statement template table # 8 – predefined subline group detail.
4. **TEMPTMPLT1** to **GLFSTMPLT1** – Financial statement template table # 1 – top-level header.
5. **TEMPTMPLT2** to **GLFSTMPLT2** – Financial statement template table # 2 – variant header.
6. **TEMPTMPLT3** to **GLFSTMPLT3** – Financial statement template table # 3 – column definitions.
7. **TEMPTMPLT4** to **GLFSTMPLT4** – Financial statement template table # 4 – statement line header.
8. **TEMPTMPLT9** to **GLFSTMPLT9** – Financial statement template table # 9 – export label list for enforcing uniqueness.
9. **TEMPTMPLT5** to **GLFSTMPLT5** – Financial statement template table # 5 – statement line calculation detail.

Column F, **Encoding Instruction # (Below):**

Refers to the number of an instruction, shown below in **Program 1 Encoding Instructions**, which will describe how to set the values in **TMPLTMVRID** and **TEMPTMPLT1** through **TEMPTMPLT9**. Encoding Instructions will be performed in Instruction # order, as called from the pseudo-code.

Column G, **Decoding Instruction # (Below):**

Refers to the number of an instruction, shown below in **Program 2 Decoding Instructions**, which will describe how to convert the values in **TMPLTMVRID** and **TMPLTCOA** and use them to populate **GLFSTMPLT1** through **GLFSTMPLT9**. Decoding Instructions will be performed in Instruction # order, as called from the pseudo-code.

**Program 1 Encoding Instructions** in the source org**:**

1. **Preliminaries** (Not shown in Table 22)**:**
   1. We will populate the flat-file equivalents of **GLFSTMPLT1-9** as **TEMPTEMPLT1-9** records.
   2. We will populate the flat-file equivalent of **GLACCT** as **TEMPLTCOA** records.
2. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT6**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT6\_CMPNO**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT6\_CMPNO**, **TMPLTMVRID\_NAME** = **GLFSTMPLT6\_CMPNO\_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
3. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT6**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT6\_FROMYEAR**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT6\_FROMYEAR**, **TMPLTMVRID\_NAME** = **GLFSTMPLT6\_FROMYEAR\_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
4. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT6**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT6\_THRUYEAR**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT6\_THRUYEAR**, **TMPLTMVRID\_NAME** = **GLFSTMPLT6\_THRUYEAR\_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
5. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT7**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT7\_GLCMP**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT7\_GLCMP**, **TMPLTMVRID\_NAME** = **GLFSTMPLT7\_GLCMP\_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
6. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT8**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT8\_GLFSTMPLT7**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT8\_GLFSTMPLT7**, **TMPLTMVRID\_NAME** = **GLFSTMPLT8\_GLFSTMPLT7\_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
7. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT8**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT8\_CMPNO**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT8\_CMPNO**, **TMPLTMVRID\_NAME** = **GLFSTMPLT8\_CMPNO\_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
8. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT8**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT8\_GLACCT**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT8\_GLACCT**, **TMPLTMVRID\_NAME** = **GLFSTMPLT8\_GLACCT\_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
9. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT1**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT1\_CMPNO**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT1\_CMPNO**, **TMPLTMVRID\_NAME** = **GLFSTMPLT1\_CMPNO\_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
10. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT1**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT1\_DIV**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT1\_DIV**, **TMPLTMVRID\_NAME** = **GLFSTMPLT1\_DIV\_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
11. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT1**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT1\_PROJ**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT1\_PROJ**, **TMPLTMVRID\_NAME** = **GLFSTMPLT1\_PROJ\_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
12. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT2**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT2\_ GLFSTMPLT1**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT2\_ GLFSTMPLT1**, **TMPLTMVRID\_NAME** = **GLFSTMPLT2\_ GLFSTMPLT1\_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
13. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT2**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT2\_DIV**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT2\_DIV**, **TMPLTMVRID\_NAME** = **GLFSTMPLT2\_DIV\_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
14. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT2**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT2\_PROJ**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT2\_PROJ**, **TMPLTMVRID\_NAME** = **GLFSTMPLT2\_PROJ\_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
15. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT3**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT3\_ GLFSTMPLT1**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT3\_ GLFSTMPLT2**, **TMPLTMVRID\_NAME** = **GLFSTMPLT3\_ GLFSTMPLT1\_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
16. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT3**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT3\_DIV**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT3\_DIV**, **TMPLTMVRID\_NAME** = **GLFSTMPLT3\_DIV\_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
17. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT3**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT3\_PROJ**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT3\_PROJ**, **TMPLTMVRID\_NAME** = **GLFSTMPLT3\_PROJ\_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
18. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT4**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT4\_ GLFSTMPLT1**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT4\_ GLFSTMPLT2**, **TMPLTMVRID\_NAME** = **GLFSTMPLT4\_ GLFSTMPLT1\_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
19. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT4**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT4\_DIV**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT4\_DIV**, **TMPLTMVRID\_NAME** = **GLFSTMPLT4\_DIV\_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
20. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT4**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT4\_PROJ**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT4\_PROJ**, **TMPLTMVRID\_NAME** = **GLFSTMPLT4\_PROJ\_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
21. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT9**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT9\_GLCMP**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT9\_GLCMP**, **TMPLTMVRID\_NAME** = **GLFSTMPLT9\_GLCMP\_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
22. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT9**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT9\_GLFSTMPLT4**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT9\_GLFSTMPLT4**, **TMPLTMVRID\_NAME** = **GLFSTMPLT9\_GLFSTMPLT4\_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
23. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT5**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT5\_GLFSTMPLT4**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT5\_GLFSTMPLT4**, **TMPLTMVRID\_NAME** = **GLFSTMPLT5\_GLFSTMPLT4\_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
24. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT5**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT5\_CMPNO**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT5\_CMPNO**, **TMPLTMVRID\_NAME** = **GLFSTMPLT5\_CMPNO \_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
25. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT5**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT5\_GLACCT**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT5\_GLACCT**, **TMPLTMVRID\_NAME** = **GLFSTMPLT5\_GLACCT \_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.
26. Insert record into **TMPLTMVRID** where **TMPLTMVRID\_TABLE** = “**GLFSTMPLT5**”, **TMPLTMVRID\_FIELDNAME** = “**GLFSTMPLT5\_GLFSTMPLT7**”, **TMPLTMVRID\_SRCID** = **GLFSTMPLT5\_GLFSTMPLT7**, **TMPLTMVRID\_NAME** = **GLFSTMPLT5\_GLACCT \_\_r.NAME**, and **TMPLTMVRID\_DESTID** = null.

**Program 2 Decoding Instructions** in the destination org**:**

1. **Preliminaries:**
   * + - 1. We will clear out and populate temp tables **TEMPTMPLT1-9**, **TMPLTMVRID**, and **TMPLTCOA** from their respective flat file(s).
         2. We will build a list of type **TMPLTMVRID\_\_c** called **TMPLTMVRIDlist** and will populate it from the **TMPLTMVRID** table built from the imported flat file(s). We will define but not populate a map called **SrcToDestMap** having a string key and string return data.
         3. We will build a list of type **GLCMP\_\_c** called **GLCMPlist** and a map called **GLCMPnameMap** whose key is **GLCMP.Name** and whose data is the record ID for the row in **GLCMP** in the destination org. Then:

We will loop through **TMPLTMVRIDlist**. For each record:

We will do a Get to **GLCMPnameMap** using **TMPLTMVRID\_NAME** as the key.

If the record ID returned is not null:

We will set the current **TMPLTMVRIDlist** record’s **TMPLTMVRID\_DESTID** to the record ID returned.

We will do a Put to **SrcToDestMap** with the current **TMPLTMVRIDlist** record’s **TMPLTMVRID\_SRCID** as the key and the record ID returned as the data.

* + - * 1. We will build a list of type **GLYEAR\_\_c** called **GLYEARlist** and a map called **GLYEARnameMap** whose key is **GLYEAR.Name** and whose data is the record ID for the row in **GLYEAR** in the destination org. Then:

We will loop through **TMPLTMVRIDlist**. For each record:

We will do a Get to **GLYEARnameMap** using **TMPLTMVRID\_NAME** as the key.

If the record ID returned is not null:

We will set the current **TMPLTMVRIDlist** record’s **TMPLTMVRID\_DESTID** to the record ID returned.

We will do a Put to **SrcToDestMap** with the current **TMPLTMVRIDlist** record’s **TMPLTMVRID\_SRCID** as the key and the record ID returned as the data.

* + - * 1. We will build a list of type **GLDIV\_\_c** called **GLDIVlist** and a map called **GLDIVnameMap** whose key is **GLDIV.Name** and whose data is the record ID for the row in **GLDIV** in the destination org. Then:

We will loop through **TMPLTMVRIDlist**. For each record:

We will do a Get to **GLDIVnameMap** using **TMPLTMVRID\_NAME** as the key.

If the record ID returned is not null:

We will set the current **TMPLTMVRIDlist** record’s **TMPLTMVRID\_DESTID** to the record ID returned.

We will do a Put to **SrcToDestMap** with the current **TMPLTMVRIDlist** record’s **TMPLTMVRID\_SRCID** as the key and the record ID returned as the data.

* + - * 1. We will build a list of type **GLPROJ\_\_c** called **GLPROJlist** and a map called **GLPROJnameMap** whose key is **GLPROJ.Name** and whose data is the record ID for the row in **GLPROJ** in the destination org. Then:

We will loop through **TMPLTMVRIDlist**. For each record:

We will do a Get to **GLPROJnameMap** using **TMPLTMVRID\_NAME** as the key.

If the record ID returned is not null:

We will set the current **TMPLTMVRIDlist** record’s **TMPLTMVRID\_DESTID** to the record ID returned.

We will do a Put to **SrcToDestMap** with the current **TMPLTMVRIDlist** record’s **TMPLTMVRID\_SRCID** as the key and the record ID returned as the data.

* + - * 1. We will build a list of type **GLACCT\_\_c** called **GLACCTlist**. We will build a list of type **TMPLTCOA\_\_c** called **TMPLTCOAlist**.We will build a map called **TMPLTCOAnameMap** whose key is **TMPLTCOA\_GLACCTNAME** and whose data is the index of the record in **TMPLTCOAlist**. We will define a map called **TMPLTCOAidMap** whose key is **TMPLTCOA\_SRCID** and whose data is **TMPLTCOA\_DESTID**. Then:

We will loop through **GLACCTlist**. For each record:

We will find the corresponding record in **TMPLTCOAlist** by reading **TMPLTCOAnameMap** with **GLACCTlist().Name**.

We will set that record’s **TMPLTCOA\_DESTID** to **GLACCTlist().Id**.

We will do a Put to **TMPLTCOAidMap** with key = **TMPLTCOA\_SRCID** and data = the new **TMPLTCOA\_DESTID**.

1. Define a list of type **GLFSTMPLT6\_\_c** called **GLFSTMPLT6List**. Define **ThisGLFSTMPLT6** as being a single row of type **GLFSTMPLT6\_\_c**.
   * + - 1. Read all records in **TEMPTEMPLT6**. For each record:

Construct an instance of **ThisGLFSTMPLT6** with **GLFSTMPLT6\_CMPNO**, **GLFSTMPLT6\_FROMYEAR**, and **GLFSTMPLT6\_THRUYEAR** initialized to null.

Set **ThisGLFSTMPLT6.GLFSTMPLT6\_CMPNO** tothe return from **SrcToDestMap**.Get(**TEMPTEMPLT6\_CMPNO**).

Set **ThisGLFSTMPLT6.GLFSTMPLT6\_FROMYEAR** tothe return from **SrcToDestMap**.Get(**TEMPTEMPLT6\_FROMYEAR**).

Set **ThisGLFSTMPLT6.GLFSTMPLT6\_THRUYEAR** tothe return from **SrcToDestMap**.Get(**TEMPTEMPLT6\_THRUYEAR**).

Add **ThisGLFSTMPLT6** to **GLFSTMPLT6List**.

* + - * 1. Insert **GLFSTMPLT6List**.

1. Define a list of type **GLFSTMPLT7\_\_c** called **GLFSTMPLT7List**. Define **ThisGLFSTMPLT7** as being a single row of type **GLFSTMPLT7\_\_c**.
   * + - 1. Read all records in **TEMPTEMPLT7**. For each record:

Construct an instance of **ThisGLFSTMPLT7** with **GLFSTMPLT7\_GLCMP** initialized to null.

Set **ThisGLFSTMPLT6.GLFSTMPLT7\_GLCMP** tothe return from **SrcToDestMap**.Get(**TEMPTEMPLT7\_GLCMP**).

Add **ThisGLFSTMPLT7** to **GLFSTMPLT7List**.

* + - * 1. Insert **GLFSTMPLT7List**.
        2. For each **GLFSTMPLT7** inserted, get its record ID and do a **SrcToDestMap**.Put(The **TEMPTEMPLT5\_GLFSTMPLT7** corresponding to the record just inserted, The new record ID). Also do a **SrcToDestMap**.Put(The **TEMPTEMPLT8\_GLFSTMPLT7** corresponding to the record just inserted, The new record ID).

1. Define a list of type **GLFSTMPLT8\_\_c** called **GLFSTMPLT8List**. Define **ThisGLFSTMPLT8** as being a single row of type **GLFSTMPLT8\_\_c**.
   * + - 1. Read all records in **TEMPTEMPLT8**. For each record:

Construct an instance of **ThisGLFSTMPLT8** with **GLFSTMPLT8\_CMPNO** and **GLFSTMPLT8\_GLACCT** initialized to null.

Set **ThisGLFSTMPLT8.GLFSTMPLT8\_CMPNO** tothe return from **SrcToDestMap**.Get(**TEMPTEMPLT8\_CMPNO**).

Set **ThisGLFSTMPLT8.GLFSTMPLT8\_GLACCT** tothe return from **TMPLTCOAidMap**.Get(**TEMPTEMPLT8\_GLACCT**).

Add **ThisGLFSTMPLT8** to **GLFSTMPLT8List**.

* + - * 1. Insert **GLFSTMPLT8List**.

1. Define a list of type **GLFSTMPLT1\_\_c** called **GLFSTMPLT1List**. Define **ThisGLFSTMPLT1** as being a single row of type **GLFSTMPLT1\_\_c**.
   * + - 1. Read all records in **TEMPTEMPLT1**. For each record:

Construct an instance of **ThisGLFSTMPLT1** with **GLFSTMPLT1\_CMPNO**, **GLFSTMPLT1\_DIV**, and **GLFSTMPLT1\_PROJ** initialized to null.

Set **ThisGLFSTMPLT1.GLFSTMPLT1\_CMPNO** tothe return from **SrcToDestMap**.Get(**TEMPTEMPLT1\_CMPNO**).

Set **ThisGLFSTMPLT1.GLFSTMPLT1\_DIV** tothe return from **SrcToDestMap**.Get(**TEMPTEMPLT1\_DIV**).

Set **ThisGLFSTMPLT1.GLFSTMPLT1\_PROJ** tothe return from **SrcToDestMap**.Get(**TEMPTEMPLT1\_PROJ**).

Add **ThisGLFSTMPLT1** to **GLFSTMPLT1List**.

* + - * 1. Insert **GLFSTMPLT1List**.
        2. For each **GLFSTMPLT1** inserted, get its record ID and do a **SrcToDestMap**.Put(The **TEMPTEMPLT2\_GLFSTMPLT1** corresponding to the record just inserted, The new record ID).

1. Define a list of type **GLFSTMPLT2\_\_c** called **GLFSTMPLT2List**. Define **ThisGLFSTMPLT2** as being a single row of type **GLFSTMPLT2\_\_c**.
   * + - 1. Read all records in **TEMPTEMPLT2**. For each record:

Construct an instance of **ThisGLFSTMPLT2** with **GLFSTMPLT2\_GLFSTMPLT1**, **GLFSTMPLT2\_DIV**, and **GLFSTMPLT2\_PROJ** initialized to null.

Set **ThisGLFSTMPLT2.GLFSTMPLT2\_GLFSTMPLT1** tothe return from **SrcToDestMap**.Get(**TEMPTEMPLT2\_GLFSTMPLT1**).

Set **ThisGLFSTMPLT2.GLFSTMPLT2\_DIV** tothe return from **SrcToDestMap**.Get(**TEMPTEMPLT2\_DIV**).

Set **ThisGLFSTMPLT2.GLFSTMPLT2\_PROJ** tothe return from **SrcToDestMap**.Get(**TEMPTEMPLT2\_PROJ**).

Add **ThisGLFSTMPLT2** to **GLFSTMPLT2List**.

* + - * 1. Insert **GLFSTMPLT2List**.
        2. For each **GLFSTMPLT2** inserted, get its record ID and do a **SrcToDestMap**.Put(The **TEMPTEMPLT3\_GLFSTMPLT2** corresponding to the record just inserted, The new record ID). Also do a **SrcToDestMap**.Put(The **TEMPTEMPLT4\_GLFSTMPLT2** corresponding to the record just inserted, The new record ID).

1. Define a list of type **GLFSTMPLT3\_\_c** called **GLFSTMPLT3List**. Define **ThisGLFSTMPLT3** as being a single row of type **GLFSTMPLT3\_\_c**.
   * + - 1. Read all records in **TEMPTEMPLT3**. For each record:

Construct an instance of **ThisGLFSTMPLT3** with **GLFSTMPLT3\_GLFSTMPLT2**, **GLFSTMPLT3\_DIV**, and **GLFSTMPLT3\_PROJ** initialized to null.

Set **ThisGLFSTMPLT3.GLFSTMPLT3\_GLFSTMPLT2** tothe return from **SrcToDestMap**.Get(**TEMPTEMPLT3\_GLFSTMPLT2**).

Set **ThisGLFSTMPLT3.GLFSTMPLT3\_DIV** tothe return from **SrcToDestMap**.Get(**TEMPTEMPLT3\_DIV**).

Set **ThisGLFSTMPLT3.GLFSTMPLT3\_PROJ** tothe return from **SrcToDestMap**.Get(**TEMPTEMPLT3\_PROJ**).

Add **ThisGLFSTMPLT3** to **GLFSTMPLT3List**.

* + - * 1. Insert **GLFSTMPLT3List**.

1. Define a list of type **GLFSTMPLT4\_\_c** called **GLFSTMPLT4List**. Define **ThisGLFSTMPLT4** as being a single row of type **GLFSTMPLT4\_\_c**.
   * + - 1. Read all records in **TEMPTEMPLT4**. For each record:

Construct an instance of **ThisGLFSTMPLT4** with **GLFSTMPLT4\_GLFSTMPLT2**, **GLFSTMPLT4\_DIV**, and **GLFSTMPLT4\_PROJ** initialized to null.

Set **ThisGLFSTMPLT4.GLFSTMPLT4\_GLFSTMPLT2** tothe return from **SrcToDestMap**.Get(**TEMPTEMPLT4\_GLFSTMPLT2**).

Set **ThisGLFSTMPLT4.GLFSTMPLT4\_DIV** tothe return from **SrcToDestMap**.Get(**TEMPTEMPLT4\_DIV**).

Set **ThisGLFSTMPLT4.GLFSTMPLT4\_PROJ** tothe return from **SrcToDestMap**.Get(**TEMPTEMPLT4\_PROJ**).

Add **ThisGLFSTMPLT4** to **GLFSTMPLT4List**.

* + - * 1. Insert **GLFSTMPLT4List**.
        2. For each **GLFSTMPLT4** inserted, get its record ID and do a **SrcToDestMap**.Put(The **TEMPTEMPLT5\_GLFSTMPLT4** corresponding to the record just inserted, The new record ID). Also do a **SrcToDestMap**.Put(The **TEMPTEMPLT9\_GLFSTMPLT4** corresponding to the record just inserted, The new record ID).

1. Define a list of type **GLFSTMPLT9\_\_c** called **GLFSTMPLT9List**. Define **ThisGLFSTMPLT9** as being a single row of type **GLFSTMPLT9\_\_c**.
   * + - 1. Read all records in **TEMPTEMPLT9**. For each record:

Construct an instance of **ThisGLFSTMPLT9** with **GLFSTMPLT9\_GLFSTMPLT4** and **GLFSTMPLT9\_GLCMP** initialized to null.

Set **ThisGLFSTMPLT9.GLFSTMPLT9\_GLFSTMPLT4** tothe return from **SrcToDestMap**.Get(**TEMPTEMPLT9\_GLFSTMPLT4**).

Set **ThisGLFSTMPLT9.GLFSTMPLT9\_GLCMP** tothe return from **SrcToDestMap**.Get(**TEMPTEMPLT9\_GLCMP**).

Add **ThisGLFSTMPLT9** to **GLFSTMPLT9List**.

* + - * 1. Insert **GLFSTMPLT9List**.

1. Define a list of type **GLFSTMPLT5\_\_c** called **GLFSTMPLT5List**. Define **ThisGLFSTMPLT5** as being a single row of type **GLFSTMPLT5\_\_c**.
   * + - 1. Read all records in **TEMPTEMPLT5**. For each record:

Construct an instance of **ThisGLFSTMPLT5** with **GLFSTMPLT5\_GLFSTMPLT4**, **GLFSTMPLT5\_CMPNO**, **GLFSTMPLT5\_GLACCT**, and **GLFSTMPLT5\_GLFSTMPLT7** initialized to null.

Set **ThisGLFSTMPLT5.GLFSTMPLT5\_GLFSTMPLT4** tothe return from **SrcToDestMap**.Get(**TEMPTEMPLT5\_GLFSTMPLT4**).

Set **ThisGLFSTMPLT5.GLFSTMPLT5\_CMPNO** tothe return from **SrcToDestMap**.Get(**TEMPTEMPLT5\_CMPNO**).

Set **ThisGLFSTMPLT5.GLFSTMPLT5\_GLACCT** tothe return from **TMPLTCOAidMap**.Get(**TEMPTEMPLT5\_GLACCT**).

Set **ThisGLFSTMPLT5.GLFSTMPLT5\_GLFSTMPLT7** tothe return from **SrcToDestMap**.Get(**TEMPTEMPLT5\_GLFSTMPLT7**).

Add **ThisGLFSTMPLT5** to **GLFSTMPLT5List**.

* + - * 1. Insert **GLFSTMPLT5List**.

**Pseudo-code**

**Program 1 – Outbound side**

**Prog 1 Main Loop**

**Prog 1 Initialize()**;

Display **Screen 1** (page 69);

Disable entry of **ThisTMPLTMVR.TMPLTMVR\_GLFSTMPLT1SRCNAME**;

While (true) {

If (User presses **Quit**) {

Break;

} // End if

If (User presses **Execute**) {

**Prog 1 Execute()**;

Break;

} // End if

If (User changes value of **TMPLTMVR\_ONETMPLTORALL** to All) {

Enable entry of **ThisTMPLTMVR.TMPLTMVR\_GLFSTMPLT1SRCNAME**;

Redisplay **Screen 1**;

} // End if

If (User changes value of **TMPLTMVR\_ONETMPLTORALL** to One) {

Set **ThisTMPLTMVR.TMPLTMVR\_GLFSTMPLT** to null;

Disable entry of **ThisTMPLTMVR.TMPLTMVR\_GLFSTMPLT1SRCNAME**;

Redisplay **Screen 1**;

} // End if

If (User changes value of **TMPLTMVR\_GLCMPNSRCNAME**) {

// Note: a dropdown of all **GLCMP.Name** fields is available to pick from.

Set **ThisTMPLTMVR.TMPLTMVR\_GLCMP** to **ID** of selected company.

If (**ThisTMPLTMVR.TMPLTMVR\_GLFSTMPLT1SRCNAME** is not null) {

Set **ThisTMPLTMVR.TMPLTMVR\_GLFSTMPLT** to null;

Redisplay **Screen 1**;

} // End if

} // End if

If (User changes value of **ThisTMPLTMVR.TMPLTMVR\_GLFSTMPLT1SRCNAME**) {

// Note: a dropdown of all **GLFSTMPLT2.Name** fields for **TMPLTMVR\_GLCMPNSRCNAME** is available to pick

// from.

Redisplay **Screen 1**;

} // End if

If (User changes value of **ThisTMPLTMVR.TMPLTMVR\_SUBLINES**) {

Redisplay **Screen 1**;

} // End if

} // End while

Return to Site Map;

**End of Prog 1 Main Loop**

**Prog 1 Execute()**

If (**ThisTMPLTMVR.TMPLTMVR\_ONETMPLTORALL** == One) {

**GLFSTMPLT1list(0)** = [ Select All fields

From **GLFSTMPLT1**

Where **GLFSTMPLT1\_CMPNO** = :**ThisTMPLTMVR.TMPLTMVR\_GLCMP**

And **GLFSTMPLT1\_TEMPLATE** = :**ThisTMPLTMVR.TMPLTMVR\_GLFSTMPLT1SRCNAME** ];

**GLFSTMPLT2list** = [ Select All fields

From **GLFSTMPLT2**

Where **GLFSTMPLT2\_CMPNO** = :**GLFSTMPLT1list(0).Id** ];

} else {

**GLFSTMPLT1list** = [ Select All fields

From **GLFSTMPLT1**

Where **GLFSTMPLT1\_CMPNO** = :**ThisTMPLTMVR.TMPLTMVR\_GLCMP** ];

**GLFSTMPLT2list** = [ Select All fields

From **GLFSTMPLT2**

Where **GLFSTMPLT2\_CMPNO** = :**GLFSTMPLT1list(0).Id** ];

}

**GLFSTMPLT3list** = [ Select All fields

From **GLFSTMPLT3**

Where **GLFSTMPLT3\_GLFSTMPLT2** in (:**GLFSTMPLT2list**)];

**GLFSTMPLT4list** = [ Select All fields

From **GLFSTMPLT4**

Where **GLFSTMPLT4\_GLFSTMPLT2** in (:**GLFSTMPLT2list**)];

If (**ThisTMPLTMVR.TMPLTMVR\_SUBLINES**) {

**GLFSTMPLT5list** = [ Select All fields

From **GLFSTMPLT5**

Where **GLFSTMPLT5\_GLFSTMPLT4** in (:**GLFSTMPLT4list**)];

} // End if

**GLFSTMPLT6list** = [ Select All fields

From **GLFSTMPLT6**

Where **GLFSTMPLT6\_CMPNO** = :**ThisTMPLTMVR.TMPLTMVR\_GLCMP** ];

If (**ThisTMPLTMVR.TMPLTMVR\_SUBLINES**) {

**GLFSTMPLT7list** = [ Select All fields

From **GLFSTMPLT7**

Where **GLFSTMPLT7\_CMPNO** = :**ThisTMPLTMVR.TMPLTMVR\_GLCMP** ];

**GLFSTMPLT8list** = [ Select All fields

From **GLFSTMPLT8**

Where **GLFSTMPLT8\_GLFSTMPLT7** in (:**GLFSTMPLT7list**)];

}

**GLFSTMPLT9list** = [ Select All fields

From **GLFSTMPLT9**

Where **GLFSTMPLT9\_GLFSTMPLT4** in (:**GLFSTMPLT4list**)];

// Populate **TEMPTMPLT1** list

For (Integer **i** = 0; **i** < **GLFSTMPLT1list**.Size(); **i**++) {

Copy each field in **GLFSTMPLT1list(i)** to equivalent field in **ThisTEMPTMPLT1**;

**TEMPTMPLT1List**.Add(**ThisTEMPTMPLT1**);

} // End for

// Populate **TEMPTMPLT2** list

For (Integer **i** = 0; **i** < **GLFSTMPLT2list**.Size(); **i**++) {

Copy each field in **GLFSTMPLT2list(i)** to equivalent field in **ThisTEMPTMPLT2**;

**TEMPTMPLT2List**.Add(**ThisTEMPTMPLT2**);

} // End for

// Populate **TEMPTMPLT3** list

For (Integer **i** = 0; **i** < **GLFSTMPLT3list**.Size(); **i**++) {

Copy each field in **GLFSTMPLT3list(i)** to equivalent field in **ThisTEMPTMPLT3**;

**TEMPTMPLT3List**.Add(**ThisTEMPTMPLT3**);

} // End for

// Populate **TEMPTMPLT4** list

For (Integer **i** = 0; **i** < **GLFSTMPLT4list**.Size(); **i**++) {

Copy each field in **GLFSTMPLT4list(i)** to equivalent field in **ThisTEMPTMPLT4**;

**TEMPTMPLT4List**.Add(**ThisTEMPTMPLT4**);

} // End for

If (**ThisTMPLTMVR.TMPLTMVR\_SUBLINES**) {

// Populate **TEMPTMPLT5** list

For (Integer **i** = 0; **i** < **GLFSTMPLT5list**.Size(); **i**++) {

Copy each field in **GLFSTMPLT5list(i)** to equivalent field in **ThisTEMPTMPLT5**;

**TEMPTMPLT5List**.Add(**ThisTEMPTMPLT5**);

} // End for

} // End if

// Populate **TEMPTMPLT6** list

For (Integer **i** = 0; **i** < **GLFSTMPLT6list**.Size(); **i**++) {

Copy each field in **GLFSTMPLT6list(i)** to equivalent field in **ThisTEMPTMPLT6**;

**TEMPTMPLT6List**.Add(**ThisTEMPTMPLT6**);

} // End for

If (**ThisTMPLTMVR.TMPLTMVR\_SUBLINES**) {

// Populate **TEMPTMPLT7** list

For (Integer **i** = 0; **i** < **GLFSTMPLT7list**.Size(); **i**++) {

Copy each field in **GLFSTMPLT7list(i)** to equivalent field in **ThisTEMPTMPLT7**;

**TEMPTMPLT7List**.Add(**ThisTEMPTMPLT7**);

} // End for

// Populate **TEMPTMPLT8** list

For (Integer **i** = 0; **i** < **GLFSTMPLT8list**.Size(); **i**++) {

Copy each field in **GLFSTMPLT8list(i)** to equivalent field in **ThisTEMPTMPLT8**;

**TEMPTMPLT8List**.Add(**ThisTEMPTMPLT8**);

} // End for

} // End if

// Populate **TEMPTMPLT9** list

For (Integer **i** = 0; **i** < **GLFSTMPLT9list**.Size(); **i**++) {

Copy each field in **GLFSTMPLT9list(i)** to equivalent field in **ThisTEMPTMPLT9**;

**TEMPTMPLT9List**.Add(**ThisTEMPTMPLT9**);

} // End for

// Encoding instruction 1 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT6list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT6”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT6\_CMPNO”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT6list(i).GLFSTMPLT6\_CMPNO**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT6list(i).GLFSTMPLT6\_CMPNO\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

// Encoding instruction 2 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT6list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT6”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT6\_FROMYEAR”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT6list(i).GLFSTMPLT6\_FROMYEAR**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT6list(i).GLFSTMPLT6\_FROMYEAR\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

// Encoding instruction 3 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT6list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT6”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT6\_THRUYEAR”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT6list(i).GLFSTMPLT6\_THRUYEAR**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT6list(i).GLFSTMPLT6\_THRUYEAR\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

If (**ThisTMPLTMVR.TMPLTMVR\_SUBLINES**) {

// Encoding instruction 4 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT7list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT7”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT7\_GLCMP”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT7list(i).GLFSTMPLT7\_GLCMP**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT7list(i).GLFSTMPLT7\_GLCMP\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

// Encoding instruction 5 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT8list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT8”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT8\_GLFSTMPLT7”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT8list(i).GLFSTMPLT8\_GLFSTMPLT7**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT8list(i).GLFSTMPLT8\_GLFSTMPLT7\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

// Encoding instruction 6 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT8list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT8”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT8\_CMPNO”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT8list(i).GLFSTMPLT8\_CMPNO**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT8list(i).GLFSTMPLT8\_CMPNO\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

// Encoding instruction 7 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT8list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT8”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT8\_GLACCT”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT8list(i).GLFSTMPLT8\_GLACCT**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT8list(i).GLFSTMPLT8\_GLACCT\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

} // End if

// Encoding instruction 8 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT1list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT1”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT1\_CMPNO”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT1list(i).GLFSTMPLT1\_CMPNO**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT1list(i).GLFSTMPLT1\_CMPNO\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

// Encoding instruction 9 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT1list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT1”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT1\_DIV”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT1list(i).GLFSTMPLT1\_DIV**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT1list(i).GLFSTMPLT1\_DIV\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

// Encoding instruction 10 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT1list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT1”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT1\_PROJ”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT1list(i).GLFSTMPLT1\_PROJ**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT1list(i).GLFSTMPLT1\_PROJ\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

// Encoding instruction 11 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT2list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT2”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT2\_ GLFSTMPLT1”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT2list(i).GLFSTMPLT2\_ GLFSTMPLT1**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT2list(i).GLFSTMPLT2\_ GLFSTMPLT1\_\_r. NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

// Encoding instruction 12 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT2list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT2”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT2\_DIV”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT2list(i).GLFSTMPLT2\_DIV**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT2list(i).GLFSTMPLT2\_DIV\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

// Encoding instruction 13 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT2list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT2”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT2\_PROJ”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT2list(i).GLFSTMPLT2\_PROJ**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT2list(i).GLFSTMPLT2\_PROJ\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

// Encoding instruction 14 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT3list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT3”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT3\_GLFSTMPLT1”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT3list(i).GLFSTMPLT3\_GLFSTMPLT1**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT3list(i).GLFSTMPLT3\_GLFSTMPLT1\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

// Encoding instruction 15 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT3list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT3”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT3\_DIV”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT3list(i).GLFSTMPLT3\_DIV**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT3list(i).GLFSTMPLT3\_DIV\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

// Encoding instruction 16 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT3list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT3”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT3\_PROJ”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT3list(i).GLFSTMPLT3\_PROJ**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT3list(i).GLFSTMPLT3\_PROJ\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

// Encoding instruction 17 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT4list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT4”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT4\_GLFSTMPLT1”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT4list(i).GLFSTMPLT4\_GLFSTMPLT1**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT4list(i).GLFSTMPLT4\_GLFSTMPLT1\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

// Encoding instruction 18 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT4list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT4”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT4\_DIV”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT4list(i).GLFSTMPLT4\_DIV**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT4list(i).GLFSTMPLT4\_DIV\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

// Encoding instruction 19 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT4list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT4”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT4\_PROJ”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT4list(i).GLFSTMPLT4\_PROJ**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT4list(i).GLFSTMPLT4\_PROJ\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

// Encoding instruction 20 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT9list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT9”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT9\_GLCMP”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT9list(i).GLFSTMPLT9\_GLCMP**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT9list(i).GLFSTMPLT9\_GLCMP\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

// Encoding instruction 21 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT9list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT9”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT9\_GLFSTMPLT4”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT9list(i).GLFSTMPLT9\_GLFSTMPLT4**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT9list(i).GLFSTMPLT9\_GLFSTMPLT4\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

If (**ThisTMPLTMVR.TMPLTMVR\_SUBLINES**) {

// Encoding instruction 22 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT5list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT5”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT5\_GLFSTMPLT4”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT5list(i).GLFSTMPLT5\_GLFSTMPLT4**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT5list(i).GLFSTMPLT5\_GLFSTMPLT4\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

// Encoding instruction 23 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT5list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT5”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT5\_CMPNO”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT5list(i).GLFSTMPLT5\_CMPNO**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT5list(i).GLFSTMPLT5\_CMPNO\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

// Encoding instruction 24 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT5list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT5”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT5\_GLACCT”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT5list(i).GLFSTMPLT5\_GLACCT**;

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT5list(i).GLFSTMPLT5\_GLACCT\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

// Encoding instruction 25 (from Table 22)

For (Integer **i** = 0; **i** < **GLFSTMPLT5list**.Size(); **i**++) {

**ThisTMPLTMVRID.TMPLTMVRID\_TABLE** = “GLFSTMPLT5”;

**ThisTMPLTMVRID.TMPLTMVRID\_FIELDNAME** = “GLFSTMPLT5\_GLFSTMPLT7”;

**ThisTMPLTMVRID.TMPLTMVRID\_SRCID** = **GLFSTMPLT5list(i).GLFSTMPLT5\_GLFSTMPLT7**

**ThisTMPLTMVRID.TMPLTMVRID\_NAME** = **GLFSTMPLT5list(i).GLFSTMPLT5\_GLFSTMPLT7\_\_r.NAME**;

**ThisTMPLTMVRID.TMPLTMVRID\_DESTID** = null;

**TMPLTMVRIDlist**.Add(**ThisTMPLTMVRID**);

} // End for

} // End if

Write **TMPLTMVRIDlist** to flat file for **TMPLTMVRID**;

If (**ThisTMPLTMVR.TMPLTMVR\_SUBLINES**) {

// Now load chart of accounts into **TMPLTCOA**

**TMPLTCOAlist** = [ Select **GLACCT\_ACCT**,

**ID**,

Null

From **GLACCT**

Where **GLACCT\_CMPNO** = :**GLUSR\_LASTCMPNO** ];

Write **TMPLTCOAlist** to flat file for **TMPLTCOA**;

} // End if

// Load **TMPLTMVR** with session parameters

Write **ThisTMPLTMVR** to flat file for **TMPLTMVR**;

Display message “Tempate(s) are encoded for transport.”

Get user acknowledgment;

Return;

**End of Prog 1 Execute()**

**Prog 1 Initialize()**

**TMPLTMVR\_\_c ThisTMPLTMVR**;

**ThisTMPLTMVR.TMPLTMVR\_ONETMPLTORALL**= All;

**ThisTMPLTMVR.TMPLTMVR\_SUBLINES** = true;

**ThisTMPLTMVR.TMPLTMVR\_GLCMPNSRCNAME** = **GLCMP.Name** where **GLCMP.Id** = **GLUSR\_LASTCMPNO** and **GLUSR\_SYUSR** =

current user;

**ThisTMPLTMVR.TMPLTMVR\_GLFSTMPLT1SRCNAME** = null;

List<**TMPLTMVRID\_\_c**> **TMPLTMVRIDlist** = new List<**TMPLTMVRID\_\_c**>();

**TMPLTMVRID\_\_c ThisTMPLTMVRID**;

List<**TMPLTCOA\_\_c**> **TMPLTCOAlist** = new List<**TMPLTCOA\_\_c**>();

List<**GLFSTMPLT1\_\_c**> **GLFSTMPLT1list** = new List<**GLFSTMPLT1\_\_c**>();

List<**GLFSTMPLT2\_\_c**> **GLFSTMPLT2list** = new List<**GLFSTMPLT2\_\_c**>();

List<**GLFSTMPLT3\_\_c**> **GLFSTMPLT3list** = new List<**GLFSTMPLT3\_\_c**>();

List<**GLFSTMPLT4\_\_c**> **GLFSTMPLT4list** = new List<**GLFSTMPLT4\_\_c**>();

List<**GLFSTMPLT5\_\_c**> **GLFSTMPLT5list** = new List<**GLFSTMPLT5\_\_c**>();

List<**GLFSTMPLT6\_\_c**> **GLFSTMPLT6list** = new List<**GLFSTMPLT6\_\_c**>();

List<**GLFSTMPLT7\_\_c**> **GLFSTMPLT7list** = new List<**GLFSTMPLT7\_\_c**>();

List<**GLFSTMPLT8\_\_c**> **GLFSTMPLT8list** = new List<**GLFSTMPLT8\_\_c**>();

List<**GLFSTMPLT9\_\_c**> **GLFSTMPLT9list** = new List<**GLFSTMPLT9\_\_c**>();

List<**TEMPTMPLT1\_\_c**> **TEMPTMPLT1List** = new List< **TEMPTMPLT1\_\_c**>();

List<**TEMPTMPLT2\_\_c**> **TEMPTMPLT2List** = new List< **TEMPTMPLT2\_\_c**>();

List<**TEMPTMPLT3\_\_c**> **TEMPTMPLT3List** = new List< **TEMPTMPLT3\_\_c**>();

List<**TEMPTMPLT4\_\_c**> **TEMPTMPLT4List** = new List< **TEMPTMPLT4\_\_c**>();

List<**TEMPTMPLT5\_\_c**> **TEMPTMPLT5List** = new List< **TEMPTMPLT5\_\_c**>();

List<**TEMPTMPLT6\_\_c**> **TEMPTMPLT6List** = new List< **TEMPTMPLT6\_\_c**>();

List<**TEMPTMPLT7\_\_c**> **TEMPTMPLT7List** = new List< **TEMPTMPLT7\_\_c**>();

List<**TEMPTMPLT8\_\_c**> **TEMPTMPLT8List** = new List< **TEMPTMPLT8\_\_c**>();

List<**TEMPTMPLT9\_\_c**> **TEMPTMPLT9List** = new List< **TEMPTMPLT9\_\_c**>();

If flat file for **TMPLTMVR** exists, erase and recreate it;

If flat file for **TMPLTMVRID** exists, erase and recreate it;

If flat file for **TMPLTCOA** exists, erase and recreate it;

If flat file for **TEMPMPLT1** exists, erase and recreate it;

If flat file for **TEMPMPLT2** exists, erase and recreate it;

If flat file for **TEMPMPLT3** exists, erase and recreate it;

If flat file for **TEMPMPLT4** exists, erase and recreate it;

If flat file for **TEMPMPLT5** exists, erase and recreate it;

If flat file for **TEMPMPLT6** exists, erase and recreate it;

If flat file for **TEMPMPLT7** exists, erase and recreate it;

If flat file for **TEMPMPLT8** exists, erase and recreate it;

If flat file for **TEMPMPLT9** exists, erase and recreate it;

Return;

**End of Prog 1 Initialize()**

**End of Program 1**

**Program 2 – Inbound side**

**Prog 2 Main Loop**

Display Screen 2;

**Prog 2 Initialize()**;

// Table processing sequence 1 (see Table 22)

For (Integer **i** = 0; **i** < **TEMPTMPLT6List**.Size(); **i**++) {

Set all fields in **ThisGLFSTMPLT6** to the equivalent fields in **TEMPTMPLT6List**;

**ThisGLFSTMPLT6.GLFSTMPLT6\_CMPNO** = **SrcToDestMap**.Get(**TEMPTMPLT6List(i).TEMPTMPLT6\_CMPNO**);

If (**ThisGLFSTMPLT6.GLFSTMPLT6\_CMPNO** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT6\_CMPNO** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**ThisGLFSTMPLT6.GLFSTMPLT6\_FROMYEAR** = **SrcToDestMap**.Get(**TEMPTMPLT6List(i).TEMPTMPLT6\_FROMYEAR**);

If (**ThisGLFSTMPLT6.GLFSTMPLT6\_FROMYEAR** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT6\_FROMYEAR** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**ThisGLFSTMPLT6.GLFSTMPLT6\_THRUYEAR** = **SrcToDestMap**.Get(**TEMPTMPLT6List(i).TEMPTMPLT6\_THRUYEAR**);

If (**ThisGLFSTMPLT6.GLFSTMPLT6\_THRUYEAR** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT6\_THRUYEAR** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**GLFSTMPLT6List**.Add(**ThisGLFSTMPLT6**);

} // End for

Insert **GLFSTMPLT6List**;

If (**ThisTMPLTMVR.TMPLTMVR\_SUBLINES**) {

// Table processing sequence 2 (see Table 22)

For (Integer **i** = 0; **i** < **TEMPTMPLT7List**.Size(); **i**++) {

Set all fields in **ThisGLFSTMPLT7** to the equivalent fields in **TEMPTMPLT7List**;

**ThisGLFSTMPLT7.GLFSTMPLT7\_GLCMP** = **SrcToDestMap**.Get(**TEMPTMPLT7List(i).TEMPTMPLT7\_GLCMP**);

If (**ThisGLFSTMPLT7.GLFSTMPLT7\_GLCMP** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT7\_GLCMP** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**GLFSTMPLT7List**.Add(**ThisGLFSTMPLT7**);

} // End for

Insert **GLFSTMPLT7List**;

For (Integer **i** = 0; **i** < **GLFSTMPLT7List**.Size(); **i**++) {

For (Integer **j** = 0; **j** < **TEMPTMPLT5List**.Size(); **j**++) {

If (**TEMPTMPLT5List(j).TEMPTMPLT5\_GLFSTMPLT7** == **TEMPTMPLT7List(i).Id**) {

**SrcToDestMap**.Put(**TEMPTMPLT5List(j).TEMPTMPLT5\_GLFSTMPLT7**, **GLFSTMPLT7List(i).Id**);

} // End if

} // End for

For (Integer **j** = 0; **j** < **TEMPTMPLT8List**.Size(); **j**++) {

If (**TEMPTMPLT8List(j).TEMPTMPLT8\_GLFSTMPLT7** == **TEMPTMPLT7List(i).Id**) {

**SrcToDestMap**.Put(**TEMPTMPLT8List(j).TEMPTMPLT8\_GLFSTMPLT7**, **GLFSTMPLT7List(i).Id**);

} // End if

} // End for

} // End for

// Table processing sequence 3 (see Table 22)

For (Integer **i** = 0; **i** < **TEMPTMPLT8List**.Size(); **i**++) {

Set all fields in **ThisGLFSTMPLT8** to the equivalent fields in **TEMPTMPLT8List**;

**ThisGLFSTMPLT8.GLFSTMPLT8\_CMPNO** = **SrcToDestMap**.Get(**TEMPTMPLT8List(i).TEMPTMPLT8\_CMPNO**);

If (**ThisGLFSTMPLT8.GLFSTMPLT8\_CMPNO** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT8\_CMPNO** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**ThisGLFSTMPLT8.GLFSTMPLT8\_GLACCT** = **SrcToDestMap**.Get(**TEMPTMPLT8List(i).TEMPTMPLT8\_GLACCT**);

If (**ThisGLFSTMPLT8.GLFSTMPLT8\_GLACCT** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT8\_GLACCT** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**GLFSTMPLT8List**.Add(**ThisGLFSTMPLT8**);

} // End for

Insert **GLFSTMPLT7List**;

} // End if

// Table processing sequence 4 (see Table 22)

For (Integer **i** = 0; **i** < **TEMPTMPLT1List**.Size(); **i**++) {

Set all fields in **ThisGLFSTMPLT1** to the equivalent fields in **TEMPTMPLT1List**;

**ThisGLFSTMPLT1.GLFSTMPLT1\_CMPNO** = **SrcToDestMap**.Get(**TEMPTMPLT1List(i).TEMPTMPLT1\_CMPNO**);

If (**ThisGLFSTMPLT1.GLFSTMPLT1\_CMPNO** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT1\_CMPNO** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**ThisGLFSTMPLT1.GLFSTMPLT1\_DIV** = **SrcToDestMap**.Get(**TEMPTMPLT1List(i).TEMPTMPLT1\_DIV**);

If (**ThisGLFSTMPLT1.GLFSTMPLT1\_DIV** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT1\_DIV** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**ThisGLFSTMPLT1.GLFSTMPLT1\_PROJ** = **SrcToDestMap**.Get(**TEMPTMPLT1List(i).TEMPTMPLT1\_PROJ**);

If (**ThisGLFSTMPLT1.GLFSTMPLT1\_PROJ** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT1\_PROJ** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**GLFSTMPLT1List**.Add(**ThisGLFSTMPLT1**);

} // End for

Insert **GLFSTMPLT1List**;

// Table processing sequence 5 (see Table 22)

For (Integer **i** = 0; **i** < **TEMPTMPLT2List**.Size(); **i**++) {

Set all fields in **ThisGLFSTMPLT2** to the equivalent fields in **TEMPTMPLT2List**;

**ThisGLFSTMPLT2.GLFSTMPLT2\_GLFSTMPLT1** = **SrcToDestMap**.Get(**TEMPTMPLT2List(i).TEMPTMPLT2\_GLFSTMPLT1**);

If (**ThisGLFSTMPLT2.GLFSTMPLT2\_GLFSTMPLT1** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT2\_GLFSTMPLT1** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**ThisGLFSTMPLT2.GLFSTMPLT2\_DIV** = **SrcToDestMap**.Get(**TEMPTMPLT2List(i).TEMPTMPLT2\_DIV**);

If (**ThisGLFSTMPLT2.GLFSTMPLT2\_DIV** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT2\_DIV** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**ThisGLFSTMPLT2.GLFSTMPLT2\_PROJ** = **SrcToDestMap**.Get(**TEMPTMPLT2List(i).TEMPTMPLT2\_PROJ**);

If (**ThisGLFSTMPLT2.GLFSTMPLT2\_PROJ** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT2\_PROJ** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**GLFSTMPLT2List**.Add(**ThisGLFSTMPLT2**);

} // End for

Insert **GLFSTMPLT2List**;

For (Integer **i** = 0; **i** < **GLFSTMPLT2List**.Size(); **i**++) {

For (Integer **j** = 0; **j** < **TEMPTMPLT3List**.Size(); **j**++) {

If (**TEMPTMPLT3List(j).TEMPTMPLT3\_GLFSTMPLT2** == **TEMPTMPLT2List(i).Id**) {

**SrcToDestMap**.Put(**TEMPTMPLT3List(j).TEMPTMPLT3\_GLFSTMPLT2**, **GLFSTMPLT2List(i).Id**);

} // End if

} // End for

For (Integer **j** = 0; **j** < **TEMPTMPLT4List**.Size(); **j**++) {

If (**TEMPTMPLT4List(j).TEMPTMPLT4\_GLFSTMPLT2** == **TEMPTMPLT2List(i).Id**) {

**SrcToDestMap**.Put(**TEMPTMPLT3List(j).TEMPTMPLT4\_GLFSTMPLT2**, **GLFSTMPLT2List(i).Id**);

} // End if

} // End for

} // End for

// Table processing sequence 6 (see Table 22)

For (Integer **i** = 0; **i** < **TEMPTMPLT3List**.Size(); **i**++) {

Set all fields in **ThisGLFSTMPLT3** to the equivalent fields in **TEMPTMPLT3List**;

**ThisGLFSTMPLT3.GLFSTMPLT3\_GLFSTMPLT2** = **SrcToDestMap**.Get(**TEMPTMPLT3List(i).TEMPTMPLT3\_GLFSTMPLT2**);

If (**ThisGLFSTMPLT3.GLFSTMPLT3\_GLFSTMPLT2** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT3\_GLFSTMPLT2** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**ThisGLFSTMPLT3.GLFSTMPLT3\_DIV** = **SrcToDestMap**.Get(**TEMPTMPLT3List(i).TEMPTMPLT3\_DIV**);

If (**ThisGLFSTMPLT3.GLFSTMPLT3\_DIV** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT3\_DIV** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**ThisGLFSTMPLT3.GLFSTMPLT3\_PROJ** = **SrcToDestMap**.Get(**TEMPTMPLT3List(i).TEMPTMPLT3\_PROJ**);

If (**ThisGLFSTMPLT3.GLFSTMPLT3\_PROJ** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT3\_PROJ** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**GLFSTMPLT3List**.Add(**ThisGLFSTMPLT3**);

} // End for

Insert **GLFSTMPLT3List**;

// Table processing sequence 7 (see Table 22)

For (Integer **i** = 0; **i** < **TEMPTMPLT4List**.Size(); **i**++) {

Set all fields in **ThisGLFSTMPLT4** to the equivalent fields in **TEMPTMPLT4List**;

**ThisGLFSTMPLT4.GLFSTMPLT4\_GLFSTMPLT2** = **SrcToDestMap**.Get(**TEMPTMPLT4List(i).TEMPTMPLT4\_GLFSTMPLT2**);

If (**ThisGLFSTMPLT4.GLFSTMPLT4\_GLFSTMPLT2** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT4\_GLFSTMPLT2** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**ThisGLFSTMPLT4.GLFSTMPLT4\_DIV** = **SrcToDestMap**.Get(**TEMPTMPLT4List(i).TEMPTMPLT4\_DIV**);

If (**ThisGLFSTMPLT4.GLFSTMPLT4\_DIV** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT4\_DIV** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**ThisGLFSTMPLT4.GLFSTMPLT4\_PROJ** = **SrcToDestMap**.Get(**TEMPTMPLT4List(i).TEMPTMPLT4\_PROJ**);

If (**ThisGLFSTMPLT4.GLFSTMPLT4\_PROJ** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT4\_PROJ** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**GLFSTMPLT4List**.Add(**ThisGLFSTMPLT4**);

} // End for

Insert **GLFSTMPLT4List**;

// Table processing sequence 8 (see Table 22)

For (Integer **i** = 0; **i** < **TEMPTMPLT9List**.Size(); **i**++) {

Set all fields in **ThisGLFSTMPLT9** to the equivalent fields in **TEMPTMPLT9List**;

**ThisGLFSTMPLT9.GLFSTMPLT9\_GLFSTMPLT4** = **SrcToDestMap**.Get(**TEMPTMPLT9List(i).TEMPTMPLT9\_GLFSTMPLT4**);

If (**ThisGLFSTMPLT9.GLFSTMPLT9\_GLFSTMPLT4** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT9\_GLFSTMPLT4** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**ThisGLFSTMPLT9.GLFSTMPLT9\_GLCMP** = **SrcToDestMap**.Get(**TEMPTMPLT9List(i).TEMPTMPLT9\_GLCMP**);

If (**ThisGLFSTMPLT9.GLFSTMPLT9\_GLCMP** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT9\_GLCMP** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**GLFSTMPLT9List**.Add(**ThisGLFSTMPLT9**);

} // End for

Insert **GLFSTMPLT9List**;

If (**ThisTMPLTMVR.TMPLTMVR\_SUBLINES**) {

// Table processing sequence 8 (see Table 22)

For (Integer **i** = 0; **i** < **TEMPTMPLT5List**.Size(); **i**++) {

Set all fields in **ThisGLFSTMPLT5** to the equivalent fields in **TEMPTMPLT5List**;

**ThisGLFSTMPLT5.GLFSTMPLT5\_GLFSTMPLT4** =

**SrcToDestMap**.Get(**TEMPTMPLT5List(i).TEMPTMPLT5\_GLFSTMPLT4**);

If (**ThisGLFSTMPLT5.GLFSTMPLT5\_GLFSTMPLT4** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT5\_GLFSTMPLT4** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**ThisGLFSTMPLT5.GLFSTMPLT5\_CMPNO** = **SrcToDestMap**.Get(**TEMPTMPLT5List(i).TEMPTMPLT5\_CMPNO**);

If (**ThisGLFSTMPLT5.GLFSTMPLT5\_CMPNO** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT5\_CMPNO** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**ThisGLFSTMPLT5.GLFSTMPLT5\_GLACCT** = **SrcToDestMap**.Get(**TEMPTMPLT5List(i).TEMPTMPLT5\_GLACCT**);

If (**ThisGLFSTMPLT5.GLFSTMPLT5\_GLACCT** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT5\_GLACCT** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**ThisGLFSTMPLT5.GLFSTMPLT5\_GLFSTMPLT7** =

**SrcToDestMap**.Get(**TEMPTMPLT5List(i).TEMPTMPLT5\_GLFSTMPLT7**);

If (**ThisGLFSTMPLT5.GLFSTMPLT5\_GLFSTMPLT7** == null) {

Display message “Unable to assign a value for a **GLFSTMPLT5\_GLFSTMPLT7** field.”;  
Get user acknowledgment;

Return to menu;

} // End if

**GLFSTMPLT5List**.Add(**ThisGLFSTMPLT5**);

} // End for

Insert **GLFSTMPLT5List**;

} // End if

Return to menu;

**End of Prog 2 Main Loop**

**Prog 2 Initialize()**

String **Id**;

**TMPLTMVR\_\_c ThisTMPLTMVR**;

**ThisTMPLTMVR.TMPLTMVR\_ONETMPLTORALL**= All;

**ThisTMPLTMVR.TMPLTMVR\_SUBLINES** = true;

**ThisTMPLTMVR.TMPLTMVR\_GLCMPNSRCNAME** = **GLCMP.Name** where **GLCMP.Id** = **GLUSR\_LASTCMPNO** and **GLUSR\_SYUSR** =

current user;

**ThisTMPLTMVR.TMPLTMVR\_GLFSTMPLT1SRCNAME** = null;

List<**TMPLTMVRID\_\_c**> **TMPLTMVRIDlist** = new List<**TMPLTMVRID\_\_c**>();

**TMPLTMVRID\_\_c ThisTMPLTMVRID**;

List<**TMPLTCOA\_\_c**> **TMPLTCOAlist** = new List<**TMPLTCOA\_\_c**>();

List<**GLFSTMPLT1\_\_c**> **GLFSTMPLT1list** = new List<**GLFSTMPLT1\_\_c**>();

List<**GLFSTMPLT2\_\_c**> **GLFSTMPLT2list** = new List<**GLFSTMPLT2\_\_c**>();

List<**GLFSTMPLT3\_\_c**> **GLFSTMPLT3list** = new List<**GLFSTMPLT3\_\_c**>();

List<**GLFSTMPLT4\_\_c**> **GLFSTMPLT4list** = new List<**GLFSTMPLT4\_\_c**>();

List<**GLFSTMPLT5\_\_c**> **GLFSTMPLT5list** = new List<**GLFSTMPLT5\_\_c**>();

List<**GLFSTMPLT6\_\_c**> **GLFSTMPLT6list** = new List<**GLFSTMPLT6\_\_c**>();

List<**GLFSTMPLT7\_\_c**> **GLFSTMPLT7list** = new List<**GLFSTMPLT7\_\_c**>();

List<**GLFSTMPLT8\_\_c**> **GLFSTMPLT8list** = new List<**GLFSTMPLT8\_\_c**>();

List<**GLFSTMPLT9\_\_c**> **GLFSTMPLT9list** = new List<**GLFSTMPLT9\_\_c**>();

List<**TEMPTMPLT1\_\_c**> **TEMPTMPLT1List** = new List< **TEMPTMPLT1\_\_c**>();

List<**TEMPTMPLT2\_\_c**> **TEMPTMPLT2List** = new List< **TEMPTMPLT2\_\_c**>();

List<**TEMPTMPLT3\_\_c**> **TEMPTMPLT3List** = new List< **TEMPTMPLT3\_\_c**>();

List<**TEMPTMPLT4\_\_c**> **TEMPTMPLT4List** = new List< **TEMPTMPLT4\_\_c**>();

List<**TEMPTMPLT5\_\_c**> **TEMPTMPLT5List** = new List< **TEMPTMPLT5\_\_c**>();

List<**TEMPTMPLT6\_\_c**> **TEMPTMPLT6List** = new List< **TEMPTMPLT6\_\_c**>();

List<**TEMPTMPLT7\_\_c**> **TEMPTMPLT7List** = new List< **TEMPTMPLT7\_\_c**>();

List<**TEMPTMPLT8\_\_c**> **TEMPTMPLT8List** = new List< **TEMPTMPLT8\_\_c**>();

List<**TEMPTMPLT9\_\_c**> **TEMPTMPLT9List** = new List< **TEMPTMPLT9\_\_c**>();

List<**GLCMP\_\_c**> **GLCMPlist** = new List<**GLCMP\_\_c**>();

List<**GLYEAR\_\_c**> **GLYEARlist** = new List<**GLYEAR\_\_c**>();

List<**GLDIV\_\_c**> **GLDIVlist** = new List<**GLDIV\_\_c**>();

List<**GLPROJ\_\_c**> **GLPROJlist** = new List<**GLPROJ\_\_c**>();

List<**GLACCT\_\_c**> **GLACCTlist** = new List<**GLACCT\_\_c**>();

Map<String, String> **SrcToDestMap** = new Map<String, String>();

Map<String, String> **GLCMPnameMap** = new Map<String, String>();

Map<String, String> **GLYEARnameMap** = new Map<String, String>();

Map<String, String> **GLDIVnameMap** = new Map<String, String>();

Map<String, String> **GLPROJnameMap** = new Map<String, String>();

Map<String, String> **GLACCTnameMap** = new Map<String, String>();

Read from **TMPLTMVR** flat file to populate **ThisTMPLTMVR**. If error occurs,

Display message “Unable to read the **TMPLTMVR** flat file. Program must abort.”;

Return to menu;

Read from **TMPLTMVRID** flat file to populate **TMPLTMVRIDlist**. If error occurs,

Display message “Unable to read the **TMPLTMVRID** flat file. Program must abort.”;

Return to menu;

If (**ThisTMPLTMVR.TMPLTMVR\_SUBLINES**) {

Read from **TMPLTCOA** flat file to populate **TMPLTCOAlist**. If error occurs,

Display message “Unable to read the **TMPLTCOA** flat file. Program must abort.”;

Return to menu;

} // End if

Read from **TEMPTMPLT1** flat file to populate **TEMPTMPLT1List**. If error occurs,

Display message “Unable to read the **TEMPTMPLT1** flat file. Program must abort.”;

Return to menu;

Read from **TEMPTMPLT2** flat file to populate **TEMPTMPLT2List**. If error occurs,

Display message “Unable to read the **TEMPTMPLT2** flat file. Program must abort.”;

Return to menu;

Read from **TEMPTMPLT3** flat file to populate **TEMPTMPLT3List**. If error occurs,

Display message “Unable to read the **TEMPTMPLT3** flat file. Program must abort.”;

Return to menu;

Read from **TEMPTMPLT4** flat file to populate **TEMPTMPLT4List**. If error occurs,

Display message “Unable to read the **TEMPTMPLT4** flat file. Program must abort.”;

Return to menu;

If (**ThisTMPLTMVR.TMPLTMVR\_SUBLINES**) {

Read from **TEMPTMPLT5** flat file to populate **TEMPTMPLT5List**. If error occurs,

Display message “Unable to read the **TEMPTMPLT5** flat file. Program must abort.”;

Return to menu;

} // End if

Read from **TEMPTMPLT6** flat file to populate **TEMPTMPLT6List**. If error occurs,

Display message “Unable to read the **TEMPTMPLT6** flat file. Program must abort.”;

Return to menu;

If (**ThisTMPLTMVR.TMPLTMVR\_SUBLINES**) {

Read from **TEMPTMPLT7** flat file to populate **TEMPTMPLT7List**. If error occurs,

Display message “Unable to read the **TEMPTMPLT7** flat file. Program must abort.”;

Return to menu;

Read from **TEMPTMPLT8** flat file to populate **TEMPTMPLT8List**. If error occurs,

Display message “Unable to read the **TEMPTMPLT8** flat file. Program must abort.”;

Return to menu;

} // End if

Read from **TEMPTMPLT9** flat file to populate **TEMPTMPLT9List**. If error occurs,

Display message “Unable to read the **TEMPTMPLT9** flat file. Program must abort.”;

Return to menu;

// Populate **GLCMPlist** and **GLCMPnameMap**

**GLCMPlist** = [ Select **Name**, **Id**

From **GLCMP** ];

For (Integer **i** = 0; **i** < **GLCMPlist**.Size(); **i**++) {

**GLCMPnameMap**.Put(**GLCMPlist(i).Name**, **GLCMPlist(i).Id**);

} // End for

// Populate **GLYEARlist** and **GLYEARnameMap**

**GLYEARlist** = [ Select **Name**, **Id**

From **GLYEAR** ];

For (Integer **i** = 0; **i** < **GLYEARlist**.Size(); **i**++) {

**GLYEARnameMap**.Put(**GLYEARlist(i).Name**, **GLYEARlist(i).Id**);

} // End for

// Populate **GLDIVlist** and **GLDIVnameMap**

**GLDIVlist** = [ Select **Name**, **Id**

From **GLDIV** ];

For (Integer **i** = 0; **i** < **GLDIVlist**.Size(); **i**++) {

**GLDIVnameMap**.Put(**GLDIVlist(i).Name**, **GLDIVlist(i).Id**);

} // End for

// Populate **GLPROJlist** and **GLPROJnameMap**

**GLPROJlist** = [ Select **Name**, **Id**

From **GLPROJ** ];

For (Integer **i** = 0; **i** < **GLPROJlist**.Size(); **i**++) {

**GLPROJnameMap**.Put(**GLPROJlist(i).Name**, **GLPROJlist(i).Id**);

} // End for

If (**ThisTMPLTMVR.TMPLTMVR\_SUBLINES**) {

// Populate **GLACCTlist** and **GLACCTnameMap**

**GLACCTlist** = [ Select **Name**, **Id**

From **GLACCT** ];

For (Integer **i** = 0; **i** < **GLACCTlist**.Size(); **i**++) {

**GLACCTnameMap**.Put(**GLACCTlist(i).Name**, **GLACCTlist(i).Id**);

} // End for

} // End if

// Populate **TMPLTMVRIDlist.TMPLTMVRID\_DESTID** and **SrcToDestMap**

For (Integer **i** = 0; **i** < **TMPLTMVRIDlist**.Size(); **i**++) {

If ((**Id** = **GLCMPnameMap**.Get(**TMPLTMVRIDlist(i).TMPLTMVRID\_NAME**) != null) {

**SrcToDestMap**.Put(**TMPLTMVRID\_NAME**, **Id**);

**TMPLTMVRIDlist(i).TMPLTMVRID\_DESTID** = **Id**;

} else if ((**Id** = **GLYEARnameMap**.Get(**TMPLTMVRIDlist(i).TMPLTMVRID\_NAME**) != null) {

**SrcToDestMap**.Put(**TMPLTMVRID\_NAME**, **Id**);

**TMPLTMVRIDlist(i).TMPLTMVRID\_DESTID** = **Id**;

} else if ((**Id** = **GLDIVnameMap**.Get(**TMPLTMVRIDlist(i).TMPLTMVRID\_NAME**) != null) {

**SrcToDestMap**.Put(**TMPLTMVRID\_NAME**, **Id**);

**TMPLTMVRIDlist(i).TMPLTMVRID\_DESTID** = **Id**;

} else if ((**Id** = **GLPROJnameMap**.Get(**TMPLTMVRIDlist(i).TMPLTMVRID\_NAME**) != null) {

**SrcToDestMap**.Put(**TMPLTMVRID\_NAME**, **Id**);

**TMPLTMVRIDlist(i).TMPLTMVRID\_DESTID** = **Id**;

} else if (**ThisTMPLTMVR.TMPLTMVR\_SUBLINES**) {

if ((**Id** = **GLACCTnameMap**.Get(**TMPLTMVRIDlist(i).TMPLTMVRID\_NAME**) != null) {

**SrcToDestMap**.Put(**TMPLTMVRID\_NAME**, **Id**);

**TMPLTMVRIDlist(i).TMPLTMVRID\_DESTID** = **Id**;

} // End if

} // End if

} // End for

// Garbage collection

**GLCMPlist**.Clear();

**GLYEARlist**.Clear();

**GLDIVlist**.Clear();

**GLPROJlist**.Clear();

**GLACCTlist**.Clear();

**GLCMPnameMap**.Clear();

**GLYEARnameMap**.Clear();

**GLDIVnameMap**.Clear();

**GLPROJnameMap**.Clear();

**GLACCTnameMap**.Clear();

Return;

**End of Prog 2 Initialize()**

**End of Program 2**