

What's New in TPL Version 6.0 for Unix?

November 2005

This document lists features and corrections that are new in Version 6.0. Please let us know if you have any questions.

Additions and Changes

Note: Items are marked with **T** and/or **R** to indicate whether they apply to TPL Tables, TPL Report, or both.

- (T/R) **CSV input files** -- CSV (comma separated values) **and other types of delimited files** can now be used as input. In the codebook, you do not need to account for all fields in the data records. Instead, you can simply assign names to the fields you want to use and reference them by Field Number.
- (T/R) **CSV output files** -- Outputs can be exported as CSV files.
- (T) **Row Banking** -- allows narrow tables to be broken at the bottom of a page, or after a selected row, and "wrapped" to the top of the page so that the sections (banks) are displayed side by side. The following example has two banks per page.

Table Q1. Median family incomes for selected characteristics

	Median Income		Median Income
U.S. Total	\$25,800	South Atlantic	\$22,974
Owner	31,400	Owner	27,200
Renter	17,610	Renter	16,760
No cash rent	15,500	No cash rent	13,780
New England	30,461	South Central	22,500
Owner	37,480	Owner	28,256
Renter	21,233	Renter	14,964
No cash rent	16,100	No cash rent	11,829
Mid Atlantic	28,150	Mountain	24,452
Owner	35,230	Owner	29,178
Renter	19,415	Renter	17,051
No cash rent	18,600	No cash rent	11,000
North Central	25,438	Pacific	28,134
Owner	30,538	Owner	36,257
Renter	15,693	Renter	19,715
No cash rent	18,711	No cash rent	24,050

- ❑ (T) **Blank space can be inserted** after selected data rows without putting slashes in stub labels. The space can be specified as a fixed number of lines or a distance in inches or centimeters. Examples:

FOR TABLE 1 ROWS 1 to 500 by 10: SKIP 1 LINE AFTER ROWS;
 FOR ROWS 3, 6, 8: SKIP .5 INCHES AFTER ROWS;

- ❑ (T) Statements have been added for **built-in statistical functions** so that you do not need to enter the formulas with Computes and Post Computes.

- * **MEAN** weighted or unweighted
- * **VAR** - variance of sample (n - 1)
- * **VARP** - variance of whole population (n)
- * **STDERR** - standard error of means
- * **STDEV** - standard deviation of sample (n - 1)
- * **STDEVP** - standard deviation of whole population (n)

Example:

MEAN MEAN_INC 'Mean Income' MASK 99,999
 ON INCOME WEIGHTED BY POP_WGT;

- ❑ (T) **Ranking** -- Table rows can be ranked (sorted) based on the values in a selected data column. Different groups or rows can be ranked differently. The ranking can be descending or ascending. Optionally, a rank column can be added to the table to display the rank number for each row. Another option lets you keep only the top (or bottom) **n** rows for a particular ranking. With this option, you can request a row to display the residual.

The following table displays the 5 states with the highest median family income. The ranking is based on the values in column 1.

Table R1. Top 5 states ranked by median income

	Total	Inside metropolitan areas	Outside metropolitan areas
MARYLAND	35,639	36,050	25,008
HAWAII	35,300	37,185	24,600
ALASKA	33,624	36,958	30,766
CONNECTICUT	33,000	32,451	35,250
MASSACHUSETTS	31,741	32,100	28,310

The next example ranks the states by population (column 2). It shows the total population, the top 10 states, and a row containing the total population of the remaining states. A rank column is added to display the rank number.

Table R2. Top 10 states ranked by population

	Rank	Total	Sex of Householder	
			Male	Female
Total	278,000,809	192,256,424	85,744,385
California	1	30,884,939	21,538,493	9,346,446
New York	2	20,762,972	12,726,296	8,036,676
Texas	3	17,522,971	12,604,780	4,918,191
Pennsylvania	4	15,113,244	10,764,171	4,349,073
Florida	5	14,894,128	9,687,748	5,206,380
Illinois	6	13,105,866	8,912,058	4,193,807
Ohio	7	12,167,546	8,305,068	3,862,478
Michigan	8	10,221,786	7,214,528	3,007,258
New Jersey	9	8,478,962	5,676,989	2,801,973
North Carolina	10	7,292,239	5,158,557	2,133,682
Total for other states	127,556,156	89,667,735	37,888,420

A new **built-in footnote named NORANK** lets you choose what should be displayed in a rank column when a rank value is not applicable. The default is blank, but in the above example the NORANK footnote has been set to display three dots. Optionally, you can add a footnote text.

- ❑ (T/R) Many types of **rules can be double** rather than single, e.g.

FOR TABLE 2 COLUMN 0: DOWN RULE DOUBLE;

Table 2. Households by sex and education level of householder

	Total	Sex of Householder	
		Male	Female
Educational Attainment of Householder			
8 years or less	3,986	2,517	1,469
High school, 1 to 3 years	3,699	2,352	1,347
High school, 4 years	10,875	7,512	3,363
College, 1 to 3 years	5,059	3,554	1,505
College, 4 years	3,466	2,629	837
College, 5 or more years	2,915	2,257	658

- ❑ (T) Table requests are **no longer limited to 2 billion potential cells**.
- ❑ (T) **PostScript tables can now be scaled** so that more rows and columns will fit on a page. This gives much finer control than you can get by changing font sizes. For example,

SCALE 99.4%;

will reduce the overall table size to 99.4% of the original.
- ❑ (T/R) For PostScript output, you can now have one page marker at the top of a page and another, different marker at the bottom of the page by specifying both a PAGE MARKER and a **BOTTOM PAGE MARKER**.
- ❑ (T) **DATA RULE MARGIN** is a new variation of RULE MARGIN. With DATA RULE MARGIN, extra space can be added between the data values and the column dividers without affecting the spacing around the heading labels.
- ❑ (T) A new Format statement **MAXIMUM FOOTNOTE SYMBOL WIDTH** controls the spacing between the left edge of a table and the footnote symbols at the bottom of the table. If you have footnote symbols of different widths in the same table, you can use this statement to right-align the symbols with each other.
- ❑ (T) You can now specify **ROUND UP** or ROUND EVEN in a format request. ROUND EVEN is the default. Previously, you could only specify ROUND in the profile. ROUND applies to the entire request and should not have a FOR clause.
- ❑ (T/R) **Ranges of values can now be included in sets** in select and select-style conditional computes; e.g.

SELECT IF AGE IN (5, 23:45 87);
- ❑ (T) There is a **new warning message**. If any part of a data value is removed, e.g. decimal places, commas, \$, or footnote symbol, because it doesn't fit in the column, the warning will inform you that this happened. The output file for the job will contain messages with table and column information to help you look for the problem.

□ (T) **HTML Improvements**

- * HTML can use CSS (cascading style sheets) with style names that describe table parts.
- * Stub indentation more closely matches the table as displayed in Ted. Stub continuation and stub increment work.
- * All other label alignment and spacing options now work also. For example if a label has SPACE .3 INCHES, the space will show up in the HTML. Labels with multiple alignments will retain these alignments.
- * Label wrapping matches the table as displayed in Ted.
- * There are no rules in stubs.
- * Shading is supported for of all or parts of a table.
- * End users can scale tables in all commonly used browsers.
- * There is an Excel option for better display in Excel; this option can be used for browsers also, but the tables are not scalable in Internet Explorer.
- * Page Markers are retained.
- * Tables with stubs on the right retain the stub on the right when exported.
- * The Autosize option allows an entire table to fit in a single html page (no automatic banking or skipping to a new page because the table is too long).
- * Table cells are now aligned based on mask (decimal points line up).
- * All rules may be deleted though selected rules cannot be.

□ (T/R) **Building Codebooks**

A new program called **tpl conditions** can be used to assist in building codebooks for any type of data that can be used with TPL. It converts partial codebook sources into complete codebook sources, creating condition value lists based on the values in the data. In doing so, it saves you work in creating codebooks and also assures that the codebook source accurately describes the data. The program can also be used to update a codebook source when the data has changed in such a way that additional condition values are needed. This new program replaces the old codebook generator that could only be used with sequential fixed width data files.

Installation and Running Jobs

- ❑ (T/R) All new input and export options can be selected with prompts or command line arguments. In addition, there are several new format statements that can be used to set default preferences for printing or exporting outputs to other formats. For example, if you never want csv output, you can use the following statement to avoid being prompted about it:

CSV OUTPUT = NO;

- ❑ (T/R) **You now have the option of linking a PostScript display program** such as **pageview** or **ghostscript** to TPL for display of PostScript tables and reports. During installation, you will be prompted for your choice and a DISPLAY NAME statement will be included in your profile. You can add or change a DISPLAY NAME statement at any time after installation.
- ❑ (T/R) The following statements can now be in a format request (previously they were profile.tpl only):

PRINT OUTPUT = YES/NO/PROMPT;
PRINT TABLE = YES/NO/PROMPT;

Miscellaneous Notes/Changes

- ❑ (T/R) A 32 bit Linux version of TPL is now available for computers that have x86 compatible hardware .
- ❑ (T) Some items in table requests that used to get Note messages now get Warning messages. An example is a reference to a condition value that does not exist in the codebook.
- ❑ (T/R) **New Keywords** are: ASCENDING, CSV, DELIMITER, DESCENDING, FIELD, KEY, MEAN, PRIMARY, QUOTE, RANK, STDERR, STDEV, STDEVP, UP, VAR, VARP

PRIMARY is a keyword only in codebooks. You can continue to use this word as a variable name if you precede it with a : in the codebook. For example,
:PRIMARY control 3

- ❑ (T) **If you have any custom programs that use the outcells file, you will need to recompile them** with a new out_rec.i . The format has changed as part of removing the 2 billion limit on table cells.

Bug Corrections

- If a page count appeared in a page marker at the end of a table, it was sometimes filled in with 99qqq9 instead of the correct number.
- Round up did not always work correctly. (This correction was offered to users of Version 5.2 and provided to those who wanted it).
- In character output (Postscript = No), dot leaders were not handled correctly. In particular, more space was allowed for dots than was needed.
- In reporting on a hierarchical file, counts of records read for each level were not correct.
- In some cases, a TPL table continued to a new page a line or two before it needed to.
- When tables were exported as HTML, footnotes using Dingbats for the footnote symbol were not always displayed correctly.
- In some cases in processing data, control variable data errors crashed TPL.
- Row deletion did not work in tables with wafers.
- When one bank of a table had fewer heading levels than another, then the exported HTML might be incorrect.
- Several formatting options which should be applicable to single wafers were applied to all wafers. These included shading of wafer label, stub, and stub head and deletion of table head.
- Setting the alignment of spanner labels might not work.
- When tables were exported as HTML and then brought into Excel, footnote text did not spread across the entire spreadsheet.
- Some table requests which used computes based on sets of values failed. An example is:

```
COMPUTE SOME_STATES:  
  1 IF STATE IN ("MD", "VA");  
  0 IF OTHER;
```
- There were several problems with alignment of all or parts of labels.
- For wafer labels in data span position, the vertical line between the stub and data was the default data width instead of the down rule width for the rule.

- If some rows are underlined and then underline row is set to no for some of these rows, additional rows have their underlines removed.
- Eject after row for all rows could cause TPL to loop.
- In certain cases, for tables with deleted stubs, export to HTML failed.
- In tables with wafers, several statements including eject after row applied to all wafers instead of just the selected wafer.
- HTML export of tables with banks and text masks might not be correct.
- For character mode tables with \$ in mask, if page length automatic was selected, then extra \$'s appeared.