What's New in Version 8.0 of TPL

TPL Version 8.0 is a major upgrade to TPL. It contains several new features and enhancements to existing feature. For additional details on any of these features, see the Reference manual or Help files.

Major New Features in TPL 8.0

Character Variables

A data field which has non-numeric characters is not generally classed in the codebook as an observation variable. If the field has too many distinct values, it is not useful as a control variable. Consequently it should be classed as a character variable. Character variables can be used directly in TPL REPORT reports. They cannot be used directly in TPL TABLES tables but they can be used in Defines, Selects and Conditional Computes.

Many character variables have a structure. They can be broken into parts. For example, a date field such as 12/1/2014 can be broken into month, day, and year. An address field can be broken into house number, street, city, state, and postal code. If the subfields of the date field are the same length from record to record, then the existing **SUBSTR** function can be used to break it apart. If the subfield widths vary from record to record then the new **CHAR SPLIT** statement can be used. If any of the resulting subfields are all numeric, then you may wish to use them as observations. The new **OBS** function can be used in a COMPUTE statement to convert a numeric character field into an observation variable.

Consider a data file with DATE field values of 12/1/98 and 6/18/2006. You wish to split the DATE field into three new observation variables, MONTH, DAY, and YEAR. You can do this with:

```
CHAR SPLIT DATE: CMONTH "/" CDAY "/" CYEAR;
COMPUTE MONTH = OBS(CMONTH);
COMPUTE DAY = OBS(CDAY);
COMPUTE YEAR =
        OBS(CYEAR) + 1900 if OBS(CYEAR) < 1900;
        OBS(CYEAR) if OBS(CYEAR) >= 1900;
```

In this example, the slashes are used as subfield dividers. The CHAR SPLIT statement has a great deal of flexibility as discussed in the manual.

Export Options

Several new export options have been added and some existing ones have been improved.

Spreadsheet (xls, ods)

TPL now directly supports a form of **xls** and **ods**. ods is the current international standard for spreadsheets which is supported by all current spreadsheet programs. TPL ods export preserves much of the formatting of a TPL table while retaining the distinction between numeric and non-numeric fields and the precision of the numeric fields.

Webpage(html)

TPL support for html has been vastly improved. Webpages produced from TPL very closely resemble the original pages. There are options to use the original page size or to take advantage of the unlimited size of a webpage by expanding the page size so that page breaks are never caused by the length or width of the "paper". Page navigation bars remain at the top of a page when the page is scrolled.

Text Tables(txt)

In older versions of TPL, the command POSTSCRIPT = YES/NO; was used to produce tables in a text mode (ASCII) with a fix-width and size font or a graphics mode with varying fonts, colors, etc. Different commands were supported in the different modes. A text mode table could not be edited in Ted. In the new version of TPL, there is only one mode. Text Tables are an export option. So you can use whatever commands you want and you can edit the table in Ted. If you want a text table, you can then just export it. If you don't like the appearance, you can use Ted to continue editing it until it is what you want.

Data Table(dat)

A data table is a table which has been modified so that it can be used as data rather than for display. In older versions of TPL, it was produced by the command DATA TABLES; Since a Data Table is a form of Text Table it had the same awkwardness to work with as a Text Table. Now, instead of using the DATA TABLE command, you can just export the table as a Data Table.

New Formatting Options

The ability to control rules and spacing has been greatly expanded in the new version of TPL. Many old commands have been replaced by new, more flexible and more consistent commands. The old commands are still supported though their use is discouraged for new development.

The new commands allow you to change individual rules where before you had to modify all rules. You can change the color and thickness of rules. You can change whether cross rules go across the entire table or only the data part of the table. You can make rules double or dotted or dashed. The following is an example of a table that can be produced using the new system but not with the previous version:

Sample Table using New Rule Features

TOTAL MONEY INCOME	All							Average Income
	Total	Race of Householder			Type of Household			
		White	Black	Other	Married couple	Other family	Nonfamily household	
CONNECTICUT	334	308	24	2	177	46	111	39,122
MAINE	292	289	1	2	157	47	88	25,747
MASSACHUSETTS	1,205	1,144	38	23	650	196	359	38,052
NEW HAMPSHIRE	272	271	–	1	162	26	84	33,822
RHODE ISLAND	259	243	13	3	158	36	65	34,068
VERMONT	261	258	1	2	136	36	89	32,279
NEW JERSEY	1,142	1,012	100	30	664	160	318	39,777
NEW YORK	2,112	1,789	266	57	1,059	397	656	33,139
PENNSYLVANIA	1,375	1,255	97	23	814	176	385	31,190
ILLINOIS	1,226	1,036	158	32	703	185	338	33,395

‾ Data not available.

Additional New Features and Conveniences

❖ TPL ODBC can now access a database with more than 100 tables.

❖ TPL ODBC can now support schemas.

❖ Longer html links are now supported in format statements.

❖ In Windows, TPL now provides better control of which tables or pages are exported.

❖ TPL provides improved handing of exported tables when fonts are not available on your machine.

❖ In UNIX, TPL jobs may use the -Q argument to access Oracle when a database name is not required.

❖ In TED, data values may now be placed in text masks associated with observation variables.

❖ In Table Builder, REPLACE statement is now implemented.

❖ Using TED, metadata such as author may be placed in a new PDF

❖ In Table Builder, holding the cursor over some controls will cause help comments (tool tips) to appear.

❖ Codebook Abstract now reports on footnotes created in the codebook.

Bug Fixes

Note: When bugs are found, they are fixed and new versions of the system are made available. So many of the bugs listed below have been fixed in the Version 7 TPL system that you are using.

TPL (Windows/UNIX/LINUX)

- ❖ Use of ODBC to connect to a Sybase or ORACLE Database will sometimes fail.

- ❖ If multiple page markers apply to a table, inserted page numbering may not work.

- ❖ A long user-specified footnote symbol may cause TPL to fail.

- ❖ Use of FOOTNOTE SYMBOL = MATCH; along with mask changes may cause TPL to fail.

- ❖ TPL may fail for a table request using a rank variable with both OTHER and FIRST n.

- ❖ A define on multiple variables may have a missing label for one of its conditions.

- ❖ In a Windows script, a script line with over 50 characters before a blank causes TPL to fail.

TABLE BUILDER

- ❖ Trying to create a quantile without specifying a rank variable correctly produces an error message. When a user corrects the error, Table Builder crashes.

- ❖ Adding a new SQL connection statements may fail.

- ❖ Requests which have only their percent bases changed do not prompt to save.

- ❖ Some simple select statements fail.

- ❖ Use of a change directory command in Table Builder may cause TPL to fail.

- ❖ For rank control variables, ALL and Other are not available options.

CODEBOOK BUILDER

- ❖ A SQL SELECT statement fails under certain conditions.

- ❖ Selecting a database can change the active directory.

TED

- ❖ Initial values for variable masks may be incorrect.

- ❖ ZERO may occur in variable list.

- ❖ Incorrect format statements for percent conditions may be generated.

- ❖ Specifying HTML links in codebook may fail.

- ❖ Exporting a CSV file may fail if the table has a deleted wafer label.

- ❖ In some cases ghostscript may not correctly convert a file into a pdf.

- ❖ Replacing a label with a footnote-only label may not work correctly.

- ❖ An exported html table may have % symbols deleted from table cells.

UNIX/LINUX only

- ❖ TPL jobs run in background against a Oracle database may incorrectly handle the -q argument and so may fail.

- ❖ The TPL Conditions program may fail when run against an Oracle database which has null values.