# "A Table Geometry Manager for the Tk Toolkit"

*George A. Howlett*

AT&T Bell Laboratories
Allentown, Pennsylvania 18103

*ABSTRACT*

The table is a new geometry manager for the Tk toolkit. It arranges windows (widgets) based upon a non-uniform rectilinear grid. It has a simple and flexible layout policy capable of managing most interface designs.

# "A Table Geometry Manager for the Tk Toolkit"

*George A. Howlett*

AT&T Bell Laboratories
Allentown, Pennsylvania 18103

*ABSTRACT*

The table is a new geometry manager for the Tk toolkit. It arranges windows (widgets) based upon a non-uniform rectilinear grid. It has a simple and flexible layout policy capable of managing most interface designs.

**Introduction**

Toolkit geometry managers were conceived to simplify composition of graphical user interfaces by automatically handling the size and placement of the windows (widgets), arranging them according to the space it is given and its own layout procedures. But in practice, implementations have tended to be overly complex or inflexible while perpetuating bad graphical interface designs.

The table is a new geometry manager for the Tk toolkit. It arranges windows based upon a non-uniform rectilinear grid. It has a simple and flexible layout capable of managing most interface designs. It also has mechanisms for designating size constraints for both windows and white space.

The table geometry manager was founded on the following two observations.

- Gridding is a common tool, ubiquitous in graphical design. It is extremely useful for defining relations among graphical components. These relations include balance, symmetry, grouping, and scale.
- Any orthogonal arrangement of rectangles (windows) can be described in a non-uniform rectilinear grid. In this sense, X geometry management is much simpler than the general problem of graphical design, since it needs only to consider the horizontal and vertical relationships among its rectangular components.

**Description**

The table consists of individual rows and columns whose spacing define its layout. Windows, called slaves, are positioned in the table by specifying the row and column containing the upper-left corner of the slave. Only one slave can be specified at a single position in the table. But slaves may span multiple rows and columns.

By default, the table tries to arrange all its slaves in the minimum space required. It determines this by querying each slave window for its requested size. The maximum requested height for all slaves in a particular row is the row's normal size. Similarly, the maximum requested width of all slaves in a particular column is the column's normal size. This, in turn, specifies the table's size, which is the sum of the normal sizes of the rows and columns.

**Example**

The following example arranges a canvas, two scrollbars, and a label in a frame. Rows and columns are numbered starting

```
table .frame \
     .title 0,0 -columnspan 2 -anchor center \
     .canvas 1,0 -fill both \
     .vscroll 1,1 -fill y \
     .hscroll 2,0 -fill x
table columnconfigure .frame 1 -resize none
table rowconfigure .frame 2 -resize none
```

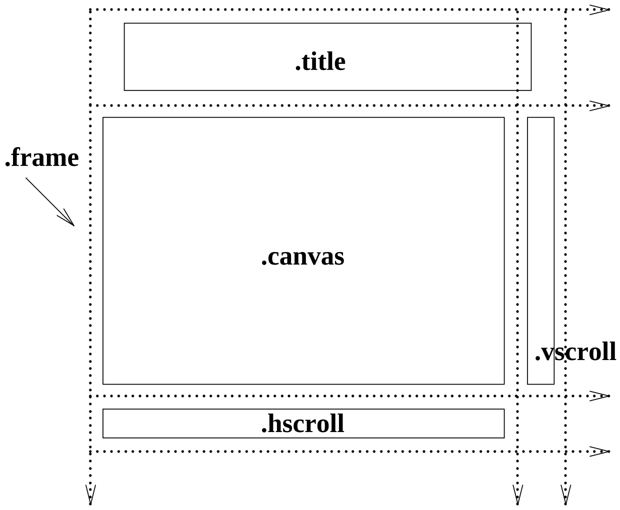


**Figure 1.** Example of Table Layout

from zero. The label `.title` spans both columns and is centered within them.

The column containing the scrollbar `.vscroll` is frozen at its normal width, and the row containing `.hscroll` is frozen at its normal height. If the frame is resized, the canvas will shrink or expand, while the scrollbars will resize only in the unconstrained directions.

**Features**

Among the features of the table geometry manager are:

**1. Simplicity**

If the layout mechanism is too cumbersome or its rules are too complicated, whatever power a geometry manager has is diminished. The arrangement of windows should be clear and predictable from the manager's description. Although TeX has a powerful stretch/shrink model, for placing graphics it is not very intuitive. On the other hand troff's tbl preprocessor is widely used, indicating that table layout is in general well-understood by users.

**2. Alignment**

Alignment is an invaluable design tool. It establishes relationships, creating strong attraction between graphic elements, even when separated by large distances. Without alignment, an interface appears unbalanced or haphazard and may suffer from spatial tension. For example, it is usually necessary to align labels and controls, but also to align across controls. A table naturally allows for both vertical and horizontal alignments.

### 3. Flattened hierarchy

The logical view of a graphical interface is flat and 2-dimensional. The table reflects this view rather than that of the X window hierarchy. Tk's pack geometry manager uses a hierarchy of windows (frames) to define rectangular regions to arrange its windows. This extra hierarchy makes it difficult to align windows both vertically and horizontally.

### 4. White space

One of the first lessons of graphical design is "white space is not wasted space". White space refers to the space not covered by the slave windows. Most geometry managers focus only on window placements, failing to consider white space at all. Any row or column of the table can be specified to have a particular size, regardless if it contains a window or not. This makes it easy to create empty areas to balance or group elements.

### 5. Constraints

The table geometry manager is the final arbiter of window size and position. By default, it considers the size requests of all the windows it manages, but it also provides mechanisms to set or constrain a window's size. For example, this makes it possible to create a rows of buttons all one inch in width, that may not be stretched greater than two inches or shrunk to less than one half inch.

In a similar fashion, the size of any row or column can be regulated. This is regardless of whether the row or column contains any windows. Constraints offer several important advantages. Many widgets do not supply adequate resources for controlling their size. In addition, it centralizes the layout description, by managing window sizes in one location.

### 6. Flexibility

The table geometry manager should be usable for the vast majority of interface designs. In contrast, Xt-based toolkits (such as Motif and OLIT) typically have many special purpose geometry managers (known as composite widgets). This works against simplicity, for the various layout rules may not be well understood or followed. And since no one of its managers is flexible enough to compose the whole interface, an extra level of hierarchy, that of multiple geometry managers, is introduced into the design process.

## Conclusion

The table is a simple and flexible geometry manager for Tk. It arranges windows in a non-uniform rectilinear grid. Its advantages are

- Offers a simple and intuitive layout.
- Aligns windows both horizontally and vertically.
- Flattens hierarchy, eliminating much of the need for extra frames to arrange windows.
- Controls sizing of both windows and white space.