A better way to add protection to a Grid row, column, or cell.

When a Grid resource (row, column or cell) has to be protected from editing, it's normal practice to
1. Assign an event procedure to the Grid (if not yet available),
2. Insert the handling of the MSG-BEGIN-ENTRY event,
3. Check if the X/Y coordinates that generated the events match with the area you wish to protect
4. If #3 above is true, make the event fail by setting the EVENT-ACTION registry

This approach is not optimized - especially in a thin client environment, as it generates client/server traffic for the event handling (the UI on the client informs the runtime on the server that an event occurred, the runtime responds that the event should fail and so the UI makes it fail).

There is a quicker and more optimized approach to protect a Grid area from editing. You can do it by setting one of the following properties to the value of "1".

- PROTECTION (to protect the whole Grid),
- ROW-PROTECTION (to protect one or more rows),
- COLUMN-PROTECTION (to protect one or more columns),
- CELL-PROTECTION (to protect one or more cells)

So, instead of code like this written in the Grid Event Procedure:

```
   evaluate event-type  when msg-begin-
entry  |first two columns are read-only       evaluate event-data-1       when 1            set event-
action to event-action-fail       when 2            set event-action to
 event-action-fail       end-evaluate  end-evaluate
```

Use code like this written in the program Procedure Division, after the DISPLAY of the Grid:

```
   |first two columns are read-only  modify screen-1-gr-1 x = 1, column-
protection = 1  modify screen-1-gr-1 x = 2, column-protection = 1
```

And your program will be more optimized.

Online URL: https://support.veryant.com/phpkb/article.php?id=275