

The DISPLAY WINDOW Statement

The DISPLAY WINDOW statement causes a window to be executed.

These statements can be coded in four divisions within each frame. These are:

WINDOW DIVISION	within the ROUTINES SECTION
PROCEDURE DIVISION	within the SECTION/ENTRY
LOAD DIVISION	
UNLOAD DIVISION	

This statement is used to display the window form, and/or to display data (i.e. a record) within this form. The window manager keeps track of the status of all windows within a frame. When a frame is first entered, clearly none of the window forms are as yet displayed. The display of the window form normally takes place automatically when it is entered, or displayed.

You may now display or update the Window Title using the following verb:

1. Statement Construct

The DISPLAY WINDOW statement causes a window or window contents to be displayed. It is coded:

```
DISPLAY WINDOW window-id [TEXT | CENTERED | TOP | [TITLE title]]
```

where *window-id* is as defined in the WINDOW statement and *title* is a PIC X field or a literal string, both of maximum length 127 bytes, containing the title to be displayed.

2. DISPLAY WINDOW *window-id*

This statement is normally used to display the data fields within a window. However, if the window has not yet been activated, the text portion of the window will first be displayed, thus activating it. The various fields defined within the window must be initialised prior to the display operation. The most frequent use of this display statement is to initialise a window prior to entering it in EDT mode, or to refresh a record that is currently displayed.

This statement provides no control over scrolled windows. If it is used to display data into a scrolled window, this data will be displayed into whichever Record Display Area happens to be current. It is therefore not possible to use this statement to display successive RDAs within a window.

While this statement can be used to display data within a window associated with a target record type, care must be taken to ensure that the I/O channel of the target record type matches with the data being displayed. The simplest way to ensure this is **not** to use the statement in such windows for any purpose other than to refresh the currently displayed record.

The display statement may be used within the routines section, and thus cause re-entrant calls on the window manager. If however any attempt is made to display a

window that is currently executing, such as when attempting to display a window from its own routines section, the frame will be aborted with a stop code.

3. **DISPLAY WINDOW *window-id* TEXT**

This statement displays the window text activating the window. It is often used to activate a number of windows at the start of a frame so that the complete, but so far empty screen is displayed prior to processing. When this is **not** done, windows are activated as required, for example when executed using the ENTER statement.

4. **DISPLAY WINDOW *window-id* TITLE *title***

This statement displays the Window displays a new title within the Window Caption Bar activating the window.

Unlike other Display Window operations, the verb may be used while the window is executing. This facility is available for use with GX only.

4. **DISPLAY WINDOW *window-id* CENTERED**

This statement is used to display the current record centred in a scrolled window. It is normally used on an already activated window following a CLEAR WINDOW *data* to redisplay the window data with the target record centred.

5. **DISPLAY WINDOW *window-id* TOP**

This statement is used to display the current record at the top of a scrolled window. It is normally used on an already activated window following a CLEAR WINDOW *data* to redisplay the window data with the target record at the top.

6. **See Also**

WINDOW statement
ENTER WINDOW statement
CLEAR WINDOW statement
WINDOW Options
WINDOW Body
ROUTINES SECTION