

# How to detect if a computer has more than one monitor and use any of them?

Today, it's common for users to work with more than one monitor.

If your application needs to take advantage multiple screens, rather than limiting windows to a single default monitor, you can use the C\$MONITOR library routine.

## Detecting the Number of Monitors

One of the primary functions of C\$MONITOR is to determine how many monitors are available to your application.

You can do this with the following call:

---

```
CALL "C$MONITOR" USING CMONITOR-GET-NO-  
M  
ONI  
TOR  
monitorCount [mainMonitor] GIVING returnCode
```

This returns the total number of monitors and optionally identifies the main monitor.

## Displaying Windows on a Specific Monitor

Once you know how many monitors are available, you can choose which one to use when displaying windows.

For example:

---

```
display standard window background-low screen line 41  
screen column 91 screen-index ws-scr-index  
... title "Screen" handle window-handle .
```

Here the SCREEN-INDEX clause lets you specify the target monitor.

## Getting Monitor Details

To position your windows precisely, you can retrieve detailed information about the monitor you want to

use.

For example, to get its size you can use this code:

---

```
CALL "C$MONITOR" USING CMONITOR-GET-MONITOR-  
IN  
FO  
  
monitorIndex          CMONITOR-DATA          GIVING returnCode
```

This call provides detailed data about the selected monitor, which you can use to fine-tune window dimensions and placement.

### Sample Program

A sample program demonstrating both of these functions is available for download. It shows how to detect monitors and display windows accordingly.

To compile the program use

---

```
iscc monitortest.cbl
```

Run it like this:

---

```
isrun MONITORTTEST
```

Online URL: <https://support.veryant.com/phpkb/article.php?id=298>