

# Detecting Media Insertion or Removal

Windows sends all top-level windows a set of default **WM\_DEVICECHANGE** messages when new devices or media (such as a CD or DVD) are added and become available, and when existing devices or media are removed. You do not need to register to receive these default messages. See the Remarks section in [RegisterDeviceNotification](#) for details on which messages are sent by default. The messages in the code example below are among the default messages.

Each **WM\_DEVICECHANGE** message has an associated event that describes the change, and a structure that provides detailed information about the change. The structure consists of an event-independent header, **DEV\_BROADCAST\_HDR**, followed by event-dependent members. The event-dependent members describe the device to which the event applies. To use this structure, applications must first determine the event type and the device type. Then, they can use the correct structure to take appropriate action.

When the user inserts a new CD or DVD into a drive, applications receive a **WM\_DEVICECHANGE** message with a **DBT\_DEVICEARRIVAL** event. The application must check the event to ensure that the type of device arriving is a volume (the **dbch\_devicetype** member is **DBT\_DEVTYP\_VOLUME**) and that the change affects the media (the **dbcv\_flags** member is **DBTF\_MEDIA**).

When the user removes a CD or DVD from a drive, applications receive a **WM\_DEVICECHANGE** message with a **DBT\_DEVICEREMOVECOMPLETE** event. Again, the application must check the event to ensure that the device being removed is a volume and that the change affects the media.

The following code demonstrates how to check for insertion or removal of a CD or DVD.

C++

```
#include <windows.h>
#include <dbt.h>
#include <strsafe.h>
#pragma comment(lib, "user32.lib" )

void Main_OnDeviceChange( HWND hwnd, WPARAM wParam, LPARAM lParam );
char FirstDriveFromMask( ULONG unitmask ); //prototype

/*-----
Main_OnDeviceChange( hwnd, wParam, lParam )

Description
Handles WM_DEVICECHANGE messages sent to the application's
top-level window.
-----*/

void Main_OnDeviceChange( HWND hwnd, WPARAM wParam, LPARAM lParam )
{
    PDEV_BROADCAST_HDR lpdb = (PDEV_BROADCAST_HDR)lParam;
    TCHAR szMsg[80];

    switch(wParam )
    {
        case DBT_DEVICEARRIVAL:
            // Check whether a CD or DVD was inserted into a drive.
            if (lpdb -> dbch_devicetype == DBT_DEVTYP_VOLUME)
            {
                PDEV_BROADCAST_VOLUME lpdbv = (PDEV_BROADCAST_VOLUME)lpdb;

                if (lpdbv -> dbcv_flags & DBTF_MEDIA)
                {
                    StringCchPrintf( szMsg, sizeof(szMsg)/sizeof(szMsg[0]),
                        TEXT("Drive %c: Media has arrived.\n"),
                        FirstDriveFromMask(lpdbv ->dbcv_unitmask) );

                    MessageBox( hwnd, szMsg, TEXT("WM_DEVICECHANGE"), MB_OK );
                }
            }
            break;

        case DBT_DEVICEREMOVECOMPLETE:
            // Check whether a CD or DVD was removed from a drive.
            if (lpdb -> dbch_devicetype == DBT_DEVTYP_VOLUME)
            {
                PDEV_BROADCAST_VOLUME lpdbv = (PDEV_BROADCAST_VOLUME)lpdb;

                if (lpdbv -> dbcv_flags & DBTF_MEDIA)
                {
                    StringCchPrintf( szMsg, sizeof(szMsg)/sizeof(szMsg[0]),
                        TEXT("Drive %c: Media was removed.\n" ),
                        FirstDriveFromMask(lpdbv ->dbcv_unitmask) );

                    MessageBox( hwnd, szMsg, TEXT("WM_DEVICECHANGE" ), MB_OK );
                }
            }
            break;

        default:
            /*
            Process other WM_DEVICECHANGE notifications for other
```

```
        devices or reasons.
    */
    ;
}
}

/*-----
FirstDriveFromMask( unitmask )

Description
    Finds the first valid drive letter from a mask of drive letters.
    The mask must be in the format bit 0 = A, bit 1 = B, bit 2 = C,
    and so on. A valid drive letter is defined when the
    corresponding bit is set to 1.

    Returns the first drive letter that was found.
-----*/

char FirstDriveFromMask( ULONG unitmask )
{
    char i;

    for (i = 0; i < 26; ++i)
    {
        if (unitmask & 0x1)
            break;
        unitmask = unitmask >> 1;
    }

    return( i + 'A' );
}
```

## Related topics

[Device Events](#)

---

## Community Additions

---