

# Monitoring the File System for Changes

---



**Jason Roberts**

.NET Developer

@robertsjason

dontcodetired.com



# Overview



## Introducing the FileSystemWatcher class

- Buffer size
- Configuring notification filters
- Additional properties

## Refactor to use a FileSystemWatcher

## Ignoring duplicate FileSystemWatcher events

- ConcurrentDictionary
- MemoryCache

## Add existing file processing

## Testing considerations



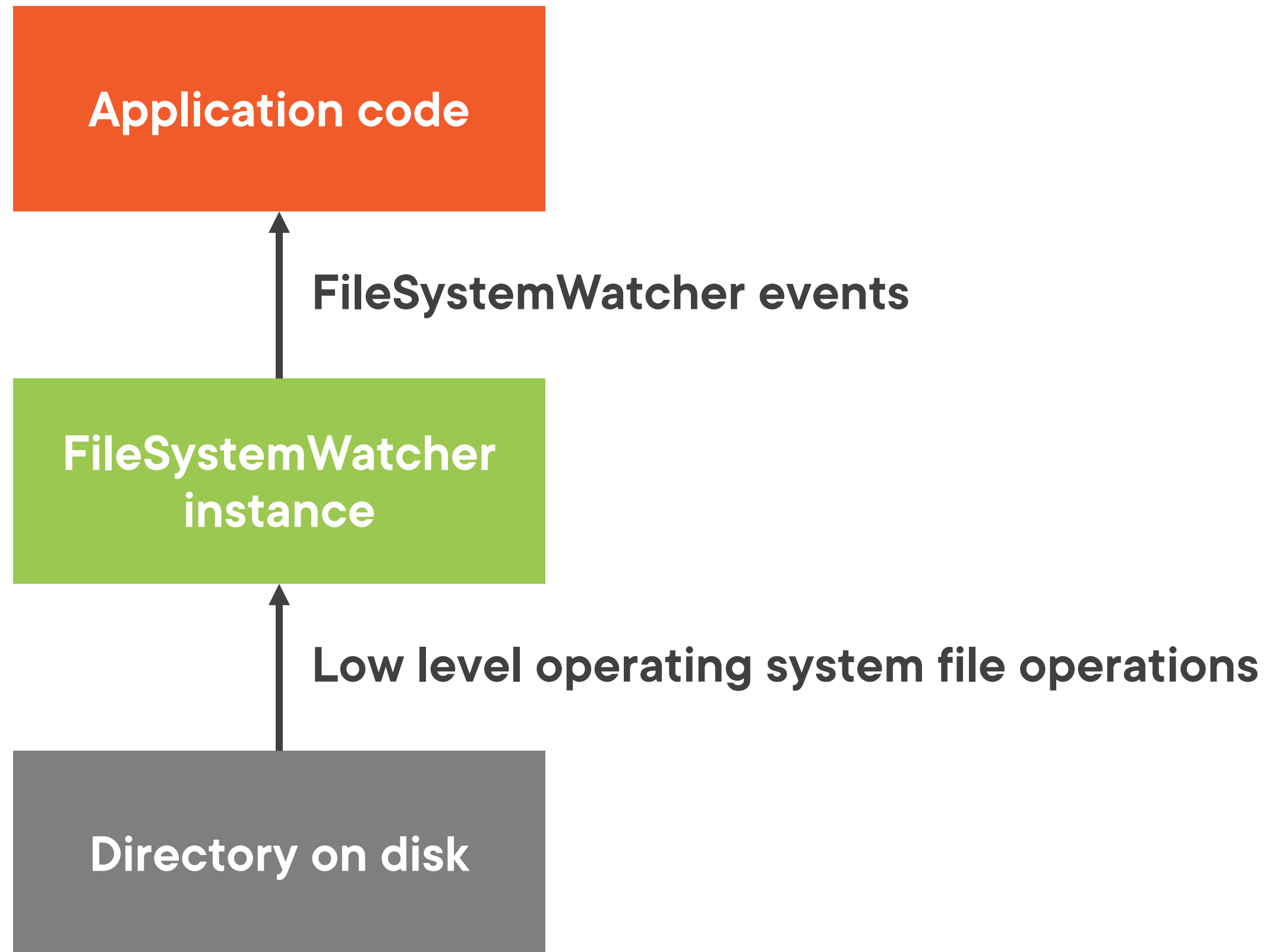
# FileSystemWatcher Class

**“Listens to the file system change notifications and raises events when a directory, or file in a directory, changes.”**

- Microsoft documentation



# Introducing the FileSystemWatcher Class



# FileSystemWatcher Events

## **Created**

- File or directory is created

## **Changed**

- File or directory is modified

## **Renamed**

- File or directory is renamed

## **Deleted**

- File or directory is deleted

## **Error**

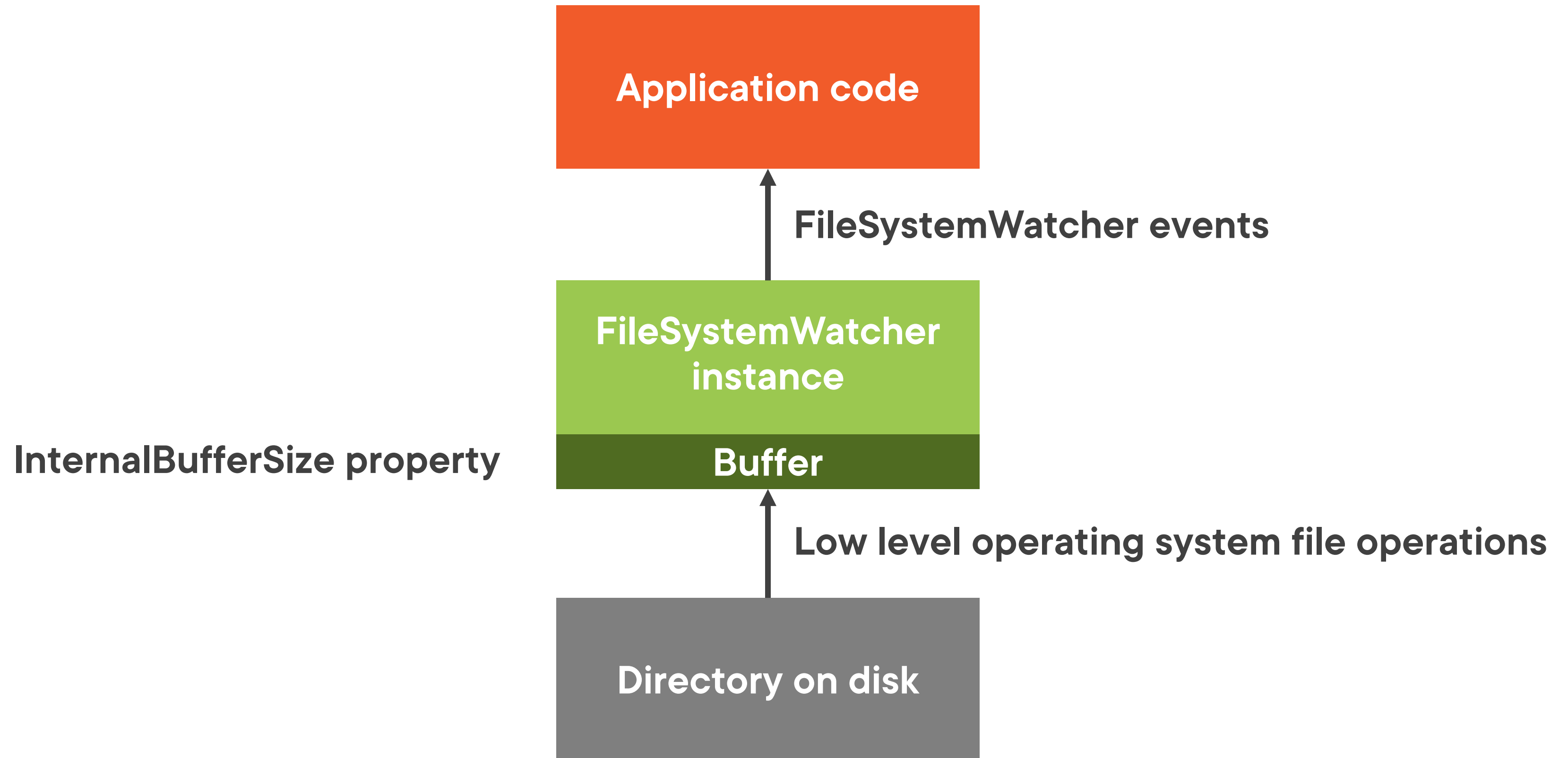
- Unable to continue monitoring changes /  
internal buffer overflow



A single file operation may  
result in multiple events being  
raised.



# InternalBufferSize Property



# InternalBufferSize Property

**Int**

**4 KB to 64 KB**

**Defaults to 8192 (8 KB)**

**16 bytes + filename**

**Multiple of 4 KB on Intel-based CPU**





“If there are many changes in a short time, the buffer can overflow. This causes the component to lose track of changes in the directory.. Increasing the size of the buffer can prevent missing file system change events. However, increasing buffer size is expensive, because it comes from non-paged memory that cannot be swapped out to disk, so keep the buffer as small as possible.”



## The NotifyFilter Property

### **NotifyFilters enum:**

- Attributes: File/directory attributes
- CreationTime: File/directory created time
- DirectoryName: The name of the directory
- FileName: The name of the file
- LastAccess: File/directory last opened date
- LastWrite: Last date file/directory written to
- Security: File/directory security settings
- Size: File/directory size

### **Combinable**



# Additional Properties

**Path**

**Filter**

**EnableRaisingEvents**



# FileSystemWatcher Testing Considerations

## Basic file operations:

- Copy
- Move
- Delete
- Overwrite
- Rename

## Incremental write of large file(s)

## Saving from different applications

## High volume of changes (buffer)

## Files only processed once

## Network/mapped drive recovery

- Error event

## Existing file processing



# Summary



## **Introduced the FileSystemWatcher class**

- InternalBufferSize property
- NotifyFilter property
- Filter property

## **Refactored to use a FileSystemWatcher**

## **Ignore duplicate FileSystemWatcher events**

- ConcurrentDictionary
- MemoryCache

## **Added existing file processing**

## **Testing considerations**



Up Next:  
Reading and Writing Entire Files  
Into Memory

---

