WatchService Interface

The final type I want to talk about, is kind of fun, it's the WatchService.

This is a special type of service, which watches registered objects for changes and events.

For example, a file manager may use a watch service, to monitor a directory for changes, so that it can update its display of the list of files, when files are created or deleted.

Using a Watch Service

A Watchable object is registered with the watch service.

When events occur on the registered object, they're queued on the watch service.

A consumer can retrieve and process the event information, using the poll or take methods on the queue.

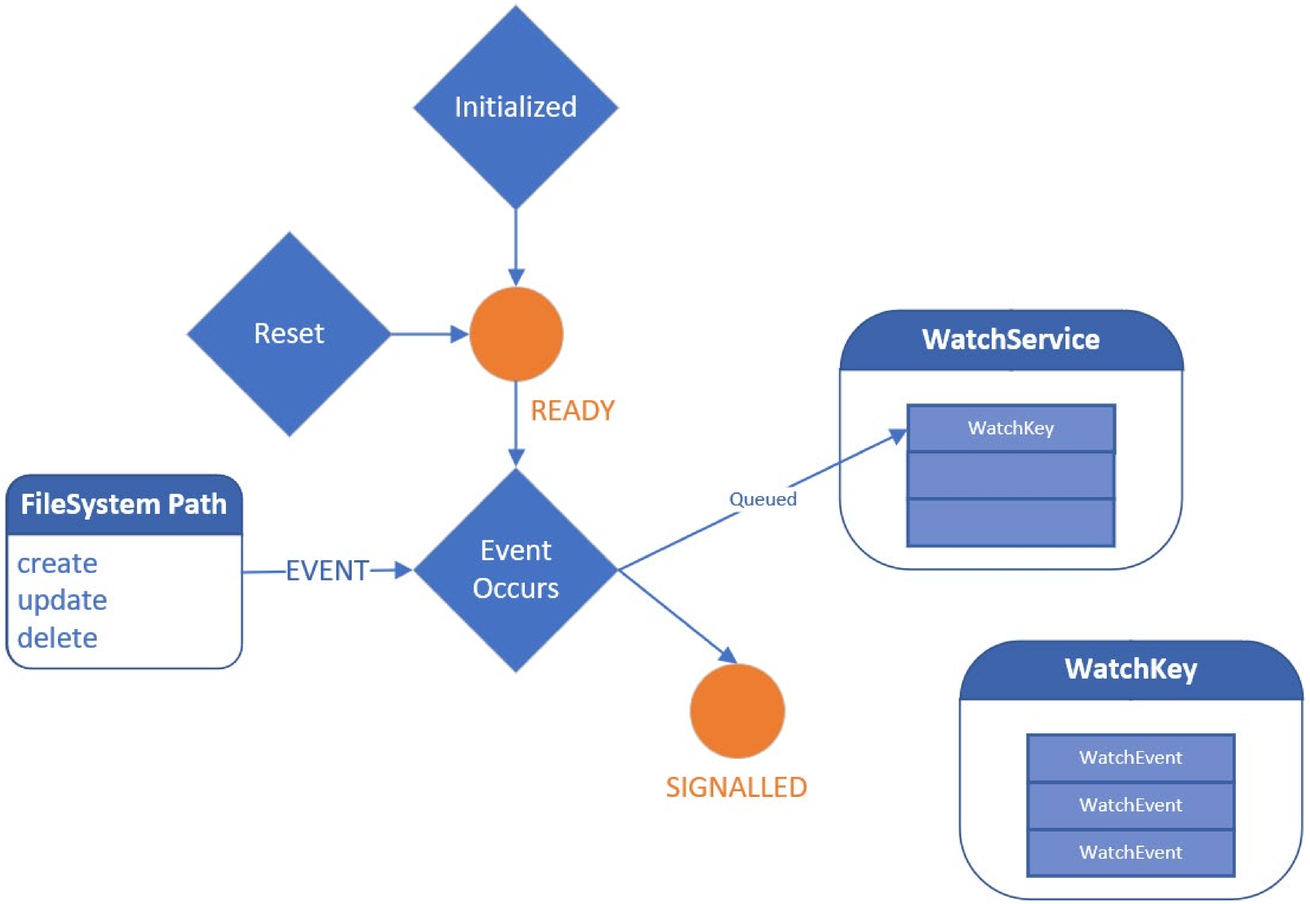
WatchKey States and Relationship

When initially created, the WatchKey is said to be **ready**.

When an event is detected, then the WatchKey is **signalled**, a special state, which means it can be polled.

A Watchkey gets queued at this point, on the WatchService, so that it can be retrieved, by invoking the watch service's poll or take methods.

Once signalled, a WatchKey remains in this state, until its reset method is invoked.



WatchKey States and Relationship

When that happens, the WatchKey returns to the ready state.

Events detected while the key is in the signalled state, are added to the WatchKey.

A Watch Key has a list events, that occurred while the WatchKey was in the signalled state.

Each WatchEvent object represents a specific change, to the watchable object.