**JMF(Java Media Framework)**

[편집하기](http://www.springnote.com/session/new?return_to=http%3A%2F%2Fhiddenviewer.springnote.com%2Fpages%2F1727246%3Fedit%3D1)

 단순한 오디오 재생기능 뿐만이 아니라, 거의 모든 종류의 멀티미디어 데이터를 처리하기 위한 기능을 수행하는 자바진영의 Media API.

1998년 중반 Sun과 SGI(Silicon Graphics), 인텔사에 의해 1.0 버전의 스펙을 디자인 했으며,

SGI와 인텔사가 개발에 불참하게 되고, 1998년 말경 IBM사가 개발에 참여 JMF 1.1버전을 발표후

현재 JMF 2.1버전까지 발표된 상태이다. Solaris용, Windows용, Cross-Platfom용 세가지 버전이 있으며,

캡처기능을 사용하고 제대로 된 기능을 이용하기 위해서는 OS에 최적화된 버전을 사용해야 한다.

JMF가 지원하는 미디어 타입

<http://java.sun.com/javase/technologies/desktop/media/jmf/2.1.1/formats.html>

1. Player player;
2. URL mediaURL = new URL("file:\c:\dear.mpg");
3. MediaLocator mrl = null;
5. if ((mrl = new MediaLocator(mediaURL)) == null) {}
7. else {
8. DataSource = null;
10. try {
11. src = Manager.createDataSource(mrl);
12. player = Manager.createPlayer(src);
14. // player를 realize시키고 Realized 상태까지 기다린다
15. player.realize();
16. while (!(processor.getState() == Processor.Realized)) {}
18. // Realized 상태에서 할일
20. player.setRate(2);
22. player를 prefetch 시키고 Prefetch상태까지 기다린다
23. player.prefetch();
24. while (!(processor.getState() == Processor.Prefetch)) {}
26. // Prefetch 상태에서 할일
28. player.syncStart();
30. } catch(IOExcetion e) {
31. } catch(javax.media.NoDataSourceException e) {
32. } catch(NoPlayerException e) {
33. }
34. }

* Clock, Duration, TimeBase

JMF에서 시간관리는 TimeBase와 Clock 두 인터페이스의 동기화를 통해 이루어진다

Clock은 미디어 시간(Media Time)을 관리하고  TimeBase는 실제시간(Time-Base Time)에 대응된다.

Media Time 0 이면 스트림의 시작을 의미하고,  최대치는 Media의 스트림이 끝나는 시점으로 Duration인테페이스를 통해 얻을 수 있다.

Media Time은 재생 중 멈추거나 다시 재생될 수 있지만, TimeBase는 실제시간이므로 한번초기화 되면 계속 흐르며 조작하는 메소드를 제공하지 않는다.

Clock의 setRate(), getRate() 메소들르 통해 재생비율을 지정하고 얻을 수 있으며

해당비율의 재생율을 지원하지 못하는 미디어에 대해서는 최대재생율로 재생하며 그값을 setRate()가 리턴한다

* DataSource
* Format에 대한 처리
* Player, Controller

* Controller의 6가지 상태

**Unrealized :** Player객체 생성시 상태. 미디어에 대한 어떠한 정보도 가지고 있지않음.

**Realizing :** Unrealized 상태에서 비동기 메소드 realize() 호출시 갖는상태. 호출 후, 바로 Realizing상태로 전이되지 않는다.

                독점적인 자원을 제외한 렌더링시에 필요한 자원을 얻는다. 독점전인 자원이란 하드웨어와 같이 오직 하나의 Player만이 사용할 수 있는 자원을 의미한다

**Realized :** realizing이 끝났을 때 갖는 상태. 미디어에 대한 정보를 알고있다. getControlPanelComponent() 메소드 호출로 미디어 재생 컨트롤 컴포넌트를 얻을 수 있고,

                미디어가 비디오를 가지고 있다면 getVisualComponent()호출로 널값이 아닌 비디오 디스플레이 컴포넌트를 얻을 수 있다.

**Prefetching :** Realized 상태에서 prefetch() 호출시 갖는상태. 미디어를 보여주기 위한 준비를 한다. 미디어 데이터를 읽어들이고, 독점적인 자원을 얻는다.

                   재생중인 미디어의 위치(시간적)가 바뀌거나, 재생비율이 바뀔때도 거치는 단계이다.

**Prefetched** : Prefetching 단계가 끝났을 때 갖는 상태로 모든 준비를 마치고 플레이 시작대기 상태

**Started** : 미디어를 플레이하고 있는 상태